X68k
Programming Series
吉野智興＋川本琢二＋山崎岳志＋実森仁志…………共著
(♯3)
X680x0
TEX

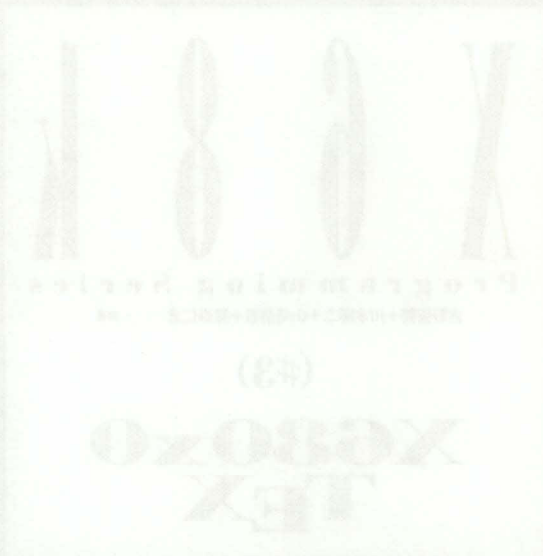
Vol.
2

Reference

SOFT
BANK

X68k
Programming Series
吉野智興＋川本琢二＋山崎岳志＋実森仁志……………共著
(♯3)
X680x0
TEX

Vol.
2

Reference

X68k Programming Series #3

TEX

Vol.2
Reference

C O N T E N T S

Chapter 1 Environments 1

1.1 ———— TEX の環境変数 …… 2
1.2 ———— FONTMAN 環境変数 …… 6
1.3 ———— PREVIEW 環境変数 …… 7
1.4 ———— PRINT 環境変数 …… 12
1.5 ———— METAFONT の環境変数 …… 17
1.6 ———— makefont の環境変数 …… 20

Chapter 2 Options 23

2.1 ———— TEX のコマンドライン …… 24
2.2 ———— FONTMAN オプション …… 25
- FONTMAN オプション概説 …… 25
- FONTMAN オプション詳説 …… 25

2.3 ———— PREVIEW オプション …… 39
- PREVIEW オプション概説 …… 39
- PREVIEW オプション詳説 …… 39

2.4 ———— PRINT オプション …… 54
- PRINT オプション概説 …… 54
- PRINT オプション詳説 …… 56
- -switch=[string] で使用できるコード …… 91

2.5 ———— METAFONT のコマンドライン …… 94
2.6 ———— makefont オプション …… 95

Chapter 3 Configuration 99

3.1 ———— コンフィギュレーションファイル文法 …… 100
- コンフィギュレーションファイルの文 …… 100
- “文” について …… 102
- “識別子” 定義 …… 103
- 文の “フォーム” …… 103
- FONTMAN のコンフィギュレーションファイル文法 …… 116

3.2 ———— p3m ファイル …… 129
- p3m ファイル概要 …… 129
- p3m ファイルフォーマット …… 130
- フォントマネージャで定義されるフォントエイリアス …… 131

Chapter 4 TEX Fonts 133

4.1 ———— 特殊記号 …… 134
- 主に文章中で利用する記号 …… 134

• 数学記号 ……135
4.2 ―― ポイント別文字サイズ比較 ……140
4.3 ―― Font Tables ……141
• 掲載フォント一覧 ……141
• フォントテーブルの参照にあたって ……143
• Font Tables ……145

Index ……177

CHAPTER 1

Environments

X680x0 版の TEX システムは、いくつかの環境変数を参照します。これらの環境変数には、UNIX 上で動作するオリジナルの TEX から受け継がれたもののほか、インストール作業などの効率化の観点から提案されたものまで含まれます。本章では、このような一連の環境変数を、TEX、フォントマネージャ、プレビューア、プリンタドライバ、METAFONT、makefont という、それぞれの実行環境ごとに説明することにしましょう。

1.1 ……………………………… TEX の環境変数

TEX は、以下の環境変数を参照します。

- TEXFORMATS　　fmt ファイルの検索パスの指定
- TEXINPUTS　　入力ファイルの検索パスの指定
- TEXFONTS　　tfm ファイルの検索パスの指定
- TEXPOOL　　ptex.pool の検索パスの指定
- TEXEDIT　　連動するエディタの指定

本書の添付ディスクのインストーラでインストールを行った場合、これらの環境変数は適切に設定されていますから、TEX の動作について十分な知識がないうちに、これらの環境変数の設定を変更するのは避けてください。トラブルの原因になります。

なお、TEX は "\" ("¥") をパスの区切りとして認識しませんので、ディレクトリを指定するためには "\" ("¥") のかわりに "/" を用いなければなりません。

TEXFORMATS　fmt ファイルの検索パスの指定

解説：　カレントディレクトリに fmt ファイルが発見できなかった場合に検索するディレクトリを指定します。“;” で区切ることで、複数のパスを指定することができます。

```
A>set TEXFORMATS=A:/MYFMT0;B:/MYFMT1/LOCAL
```

TEXINPUTS　入力ファイルの検索パスの指定

解説：　コマンドラインやコントロール・シーケンス “\input” で指定されたファイルを検索するディレクトリを指定します。“;” で区切ることで、複数のパスを指定することができます。

```
A>set TEXINPUTS=A:/MYINPUT;B:/INPUT/LOCAL
```

TEXFONTS　tfm ファイルの検索パスの指定

解説： TEX で使うフォントの情報が記述されている tfm ファイルを検索するディレクトリを指定します。“;” で区切ることで、複数のパスを指定することができます。

```
A>set TEXFONTS=A:/MYFONTS;B:/FONT/LOCAL
```

TEXPOOL　ptex.pool の検索パスの指定

解説： TEX の実行ファイルが必要とする ptex.pool を検索するディレクトリを指定します。
ptex.pool は、TEX の実行ファイルがエラーメッセージやエラーに関する詳細な情報を出力する際に必要とするファイルです。initex.x によって fmt ファイルを作成する時点で fmt ファイル内に読み込まれます。このため、環境変数 TEXPOOL の設定は initex.x の実行時にのみ必要とされますので、fmt ファイルを頻繁につくらない人は必要なときだけ指定しても実用上問題ありません。

```
A>set TEXPOOL=A:/POOLDIR
```

TEXEDIT　連動するエディタの指定

解説： TEX は、処理するソースにエラーを発見すると、対話的に作成者の指示を仰ぎますが、そのとき “e” を入力することでエディタの起動、および処理中ファイルの当該行へのジャンプを指示し、TEX を終了させることができます。

そのとき使用するエディタを、この環境変数で指定します。指定の書式は、

set TEXEDIT=エディタ名称 %s 行指定オプション %d

です。%s の部分はエラーを起こしたソースファイル名称に、%d の部分はエラーが起こった行番号に、それぞれ展開され、この 2 つの引数を「エディタ名称」で指定されたエディタに渡します。

この環境変数は、TEX 内部で C 言語の sprintf 関数にそのまま渡されます。誤った指定を行った場合には、バスエラーやアドレスエラーを引き起こしますので注意してください。TEX では安全に実行可能かどうかをチェックしていません。

```
A>set TEXEDIT=emx %s -g%d
```

1.2 ……………………………FONTMAN 環境変数

フォントマネージャで利用する環境変数は "TEXHOME" ひとつだけです。

TEXHOME　TEX の HOME ディレクトリ

解説： フォントマネージャが参照するファイルは、おおむね、この環境変数が指し示すディレクトリ内に置きます。
この環境変数の内容が、たとえば D:\tex だとすると、フォントマネージャの存在するディレクトリは D:\tex\fontman であり、ドキュメントは D:\tex\fontman\doc に入っているということになります。
なお、このような、環境変数の内容にもとづいてコンフィギュレーションファイル内で相対的なパスを記述するときは、いちいち環境変数の内容を書かずに、直接 $env[TEXHOME]\fontman とか $env[TEXHOME]\fontman\doc と書きます。

1.3 PREVIEW 環境変数

プレビューアで利用する環境変数は、以下のとおりです。

- `TEXHOME`　TeX の HOME ディレクトリ
- `TEXPKS`　TeX の pk ファイルを収めるディレクトリ
- `PREVIEW.CFG`　プレビューアのコンフィギュレーションファイル
- `PREVIEW.P2M`　FONTMAN Ver.2 への変換ファイル
- `PREVIEW.P3M`　FONTMAN Ver.3 への変換ファイル

TEXHOME TEX の HOME ディレクトリ

解説： プレビューアが参照するファイルは、おおむね、この環境変数が指し示すディレクトリ内に置きます。

この環境変数の内容が、たとえば D:\tex だとすると、プレビューアのコンフィギュレーションファイルのデフォルトは D:\tex\preview.cfg ということになりますし、プレビューアのドキュメントは D:\tex\drivers\doc に入っているということになります。

なお、このような、環境変数の内容にもとづいて相対的なパスを記述するときは、いちいち環境変数の内容を書かずに、直接 %TEXHOME%\preview.cfg や %TEXHOME%\drivers\doc と書きます。

TEXPKS TEX の pk ファイルを収めるディレクトリ

解説： pk フォントを入れたディレクトリは、デフォルトでは %TEXHOME%\fonts です。このときプレビューアは、%TEXHOME%\fonts 配下の、⟨フォント名⟩.⟨dpi⟩ pk というファイル、および dpi 値を表す 4 文字以下の数字からなるサブディレクトリ内の ⟨フォント名⟩.pk というファイルを探します。たとえば、118dpi の cmr10 の pk フォントは、%TEXHOME%\fonts\cmr10.118pk あるいは %TEXHOME%\fonts\118\cmr10.pk というフルパスでアクセスされます。

もし、これを別のパスを参照するように変更したい場合は、環境変数 TEXPKS に絶対パスをセットします。このとき、セミコロン “;” で区切って複数のパスを指定することも可能です。

PREVIEW.CFG　プレビューアのコンフィギュレーションファイル

解説： プレビューアのコンフィギュレーションファイルは、デフォルトでは %TEXHOME%\preview.cfg です。このファイル名や格納ディレクトリを変更する場合、環境変数 PREVIEW.CFG にフルパスで指定します。たとえば、ファイル名を MyPreview.cnf に変えたい場合、TeXenv.bat に List 1-1 の 1 行を追加します。

List 1-1● 環境変数 PREVIEW.CFG の設定例 (1)

```
1:  set PREVIEW.CFG=%TEXHOME%\MyPreview.cnf
```

また、格納ディレクトリを %TEXHOME% のサブディレクトリ configs に変更したい場合、TeXenv.bat に List 1-2 の 1 行を追加します。

List 1-2● 環境変数 PREVIEW.CFG の設定例 (2)

```
1:  set PREVIEW.CFG=%TEXHOME%\configs\preview.cfg
```

格納ディレクトリもファイル名も変えたい場合、TeXenv.bat に List 1-3 のような 1 行を追加します。

List 1-3● 環境変数 PREVIEW.CFG の設定例 (3)

```
1:  set PREVIEW.CFG=A:\etc\dvi2dsp.cfg
```

なお、環境変数 PREVIEW_CFG にも同じ効果がありますが、プレビューアは PREVIEW.CFG ⟶ PREVIEW_CFG の順番でファイルを参照しますので、環境変数 PREVIEW.CFG の設定が優先されます。

PREVIEW.P2M　FONTMAN Ver.2 への変換ファイル

解説： FONTMAN Ver.2[1]を利用する場合、プレビューアは preview.p2m ファイルを必要とします。このときプレビューアは、デフォルトではカレントディレクトリの preview.p2m を、もしカレントディレクトリになければ %TEXHOME% の preview.p2m を参照します。

このファイル名や格納ディレクトリを変更する場合、環境変数 PREVIEW.P2M にフルパスでこれを指定します。

たとえば、ファイル名を MyPreview.p2m に変えたい場合、TeXenv.bat に List 1-4 の 1 行を追加します。

List 1-4● 環境変数 PREVIEW.P2M の設定例 (1)

```
1: set PREVIEW.P2M=%TEXHOME%\MyPreview.p2m
```

また、格納ディレクトリを %TEXHOME% のサブディレクトリ configs に変更したい場合、TeXenv.bat に List 1-5 の 1 行を追加します。

List 1-5● 環境変数 PREVIEW.P2M の設定例 (2)

```
1: set PREVIEW.P2M=%TEXHOME%\configs\preview.p2m
```

格納ディレクトリもファイル名も変えたい場合、TeXenv.bat に List 1-6 のような 1 行を追加します。

List 1-6● 環境変数 PREVIEW.P2M の設定例 (3)

```
1: set PREVIEW.P2M=A:\etc\dvi2dsp.p2m
```

なお、環境変数 PREVIEW_P2M にも同じ効果がありますが、プレビューアは PREVIEW.P2M ⟶ PREVIEW_P2M の順に参照しますので、環境変数 PREVIEW.P2M の設定が優先されます[2]。

1)FONTMAN Ver.2 は、本書には添付されていません。

2)いうまでもありませんが、プレビューア実行時に、オプション -pk2fontman が指定された場合には、オプションの設定が優先されます。

PREVIEW.P3M　FONTMAN Ver.3 への変換ファイル

解説：FONTMAN Ver.3 を利用する場合、プレビューアは preview.p3m ファイルを必要とします。プレビューアは、デフォルトではカレントディレクトリの preview.p3m を、もしカレントディレクトリになければ %TEXHOME% の preview.p3m を参照します。
このファイル名や格納ディレクトリを変更する場合、環境変数 PREVIEW.P3M にフルパスで指定してください。
たとえば、ファイル名を MyPreview.p3m に変えたい場合、TeXenv.bat に List 1-7 の 1 行を追加します。

List 1-7● 環境変数 PREVIEW.P3M の設定例 (1)

```
1: set PREVIEW.P3M=%TEXHOME%\MyPreview.p3m
```

また、格納ディレクトリを TeX のサブディレクトリ configs に変更したい場合、TeXenv.bat に List 1-8 の 1 行を追加します。

List 1-8● 環境変数 PREVIEW.P3M の設定例 (2)

```
1: set PREVIEW.P3M=%TEXHOME%\configs\preview.p3m
```

格納ディレクトリもファイル名も変えたい場合、TeXenv.bat に List 1-9 のような 1 行を追加します。

List 1-9● 環境変数 PREVIEW.P3M の設定例 (3)

```
1: set PREVIEW.P3M=A:\etc\dvi2dsp.p3m
```

なお、環境変数 PREVIEW_P3M にも同じ効果がありますが、プレビューアは PREVIEW.P3M ⟶ PREVIEW_P3M の順に参照しますので、環境変数 PREVIEW.P3M の設定が優先されます[3)]。

3)いうまでもありませんが、プレビューア実行時に、オプション -pk3fontman が指定された場合には、オプションの設定が優先されます。

1.4 ………………………………… PRINT 環境変数

プリンタドライバで利用する環境変数は、以下のとおりです。

- TEXHOME　　　TEX の HOME ディレクトリ
- TEXPKS　　　TEX の pk ファイルを収めるディレクトリ
- PRINT.CFG　　　プリンタドライバのコンフィギュレーションファイル
- PRINT.P2M　　　FONTMAN Ver.2 への変換ファイル
- PRINT.P3M　　　FONTMAN Ver.3 への変換ファイル

TEXHOME　TeX の HOME ディレクトリ

解説： プリンタドライバが参照するファイルは、おおむね、この環境変数が指し示すディレクトリ内に置きます。

この環境変数の内容が、たとえば D:\tex だとすると、プリンタドライバのコンフィギュレーションファイルのデフォルトは D:\tex\print.cfg ということになりますし、ドキュメントは D:\tex\drivers\doc に入っているということになります。

なお、このような、環境変数の内容にもとづいて相対的なパスを記述するときは、いちいち環境変数の内容を書かずに、直接 %TEXHOME%\print.cfg や %TEXHOME%\drivers\doc と書きます。

TEXPKS　TeX の pk ファイルを収めるディレクトリ

解説： pk フォントを入れたディレクトリは、デフォルトでは %TEXHOME%\fonts です。このときプリンタドライバは、%TEXHOME%\fonts 配下の、〈フォント名〉.〈dpi〉pk というファイル、および dpi 値を表す 4 文字以下の数字からなるサブディレクトリ内の〈フォント名〉.pk というファイルを探します。たとえば、360dpi の cmr10 の pk フォントは、%TEXHOME%\fonts\cmr10.360pk あるいは %TEXHOME%\fonts\360\cmr10.pk というフルパスでアクセスされます。

もし、これを別のパスを参照するように変更したい場合は、環境変数 TEXPKS に絶対パスをセットします。このとき、セミコロン“;”で区切って複数のパスを指定することも可能です。

PRINT.CFG　プリンタドライバのコンフィギュレーションファイル

解説： プリンタドライバのコンフィギュレーションファイルは、デフォルトでは %TEXHOME%\print.cfg です。このファイル名や格納ディレクトリを変更する場合、環境変数 PRINT.CFG にフルパスで指定します。たとえば、ファイル名を MyPrint.cnf に変えたい場合、TeXenv.bat に List 1-10 の 1 行を追加します。

List 1-10 ● 環境変数 PRINT.CFG の設定例 (1)

```
1:  set PRINT.CFG=%TEXHOME%\MyPrint.cnf
```

また、格納ディレクトリを %TEXHOME% のサブディレクトリ configs に変更したい場合、TeXenv.bat に List 1-11 の 1 行を追加します。

List 1-11 ● 環境変数 PRINT.CFG の設定例 (2)

```
1:  set PRINT.CFG=%TEXHOME%\configs\print.cfg
```

格納ディレクトリもファイル名も変えたい場合、TeXenv.bat に List 1-12 のような 1 行を追加します。

List 1-12 ● 環境変数 PRINT.CFG の設定例 (3)

```
1:  set PRINT.CFG=A:\etc\dvi2dsp.cfg
```

なお、環境変数 PRINT_CFG にも同じ効果がありますが、プリンタドライバは PRINT.CFG ⟶ PRINT_CFG の順番でファイルを参照しますので、環境変数 PRINT.CFG の設定が優先されます。

PRINT.P2M　FONTMAN Ver.2 への変換ファイル

解説： FONTMAN Ver.2[1]を利用する場合、プリンタドライバは print.p2m ファイルを必要とします。このときプリンタドライバは、デフォルトではカレントディレクトリの print.p2m を、もしカレントディレクトリになければ %TEXHOME% の print.p2m を参照します。
このファイル名や格納ディレクトリを変更する場合、環境変数 PRINT.P2M にフルパスでこれを指定します。
たとえば、ファイル名を MyPrint.p2m に変えたい場合、TeXenv.bat に List 1-13 の 1 行を追加します。

List 1-13 ● 環境変数 PRINT.P2M の設定例 (1)

```
1:  set PRINT.P2M=%TEXHOME%\MyPrint.p2m
```

また、格納ディレクトリを %TEXHOME% のサブディレクトリ configs に変更したい場合、TeXenv.bat に List 1-14 の 1 行を追加します。

List 1-14 ● 環境変数 PRINT.P2M の設定例 (2)

```
1:  set PRINT.P2M=%TEXHOME%\configs\print.p2m
```

格納ディレクトリもファイル名も変えたい場合、TeXenv.bat に List 1-15 のような 1 行を追加します。

List 1-15 ● 環境変数 PRINT.P2M の設定例 (3)

```
1:  set PRINT.P2M=A:\etc\dvi2dsp.p2m
```

なお、環境変数 PRINT_P2M にも同じ効果がありますが、プリンタドライバは PRINT.P2M ⟶ PRINT_P2M の順に参照しますので、環境変数 PRINT.P2M の設定が優先されます[2]。

1) FONTMAN Ver.2 は、本書には添付されていません。

2) いうまでもありませんが、プリンタドライバ実行時に、オプション -pk2fontman が指定された場合には、オプションの設定が優先されます。

PRINT.P3M　FONTMAN Ver.3 への変換ファイル

解説： FONTMAN Ver.3 を利用する場合、プリンタドライバは print.p3m ファイルを必要とします。プリンタドライバは、デフォルトではカレントディレクトリの print.p3m を、もしカレントディレクトリになければ %TEXHOME% の print.p3m を参照します。

このファイル名や格納ディレクトリを変更する場合、環境変数 PRINT.P3M にフルパスで指定してください。

たとえば、ファイル名を MyPrint.p3m に変えたい場合、TeXenv.bat に List 1-16 の 1 行を追加します。

List 1-16 ● 環境変数 PRINT.P3M の設定例 (1)

```
1:  set PRINT.P3M=%TEXHOME%\MyPrint.p3m
```

また、格納ディレクトリを TeX のサブディレクトリ configs に変更したい場合、TeXenv.bat に List 1-17 の 1 行を追加します。

List 1-17 ● 環境変数 PRINT.P3M の設定例 (2)

```
1:  set PRINT.P3M=%TEXHOME%\configs\print.p3m
```

格納ディレクトリもファイル名も変えたい場合、TeXenv.bat に List 1-18 のような 1 行を追加します。

List 1-18 ● 環境変数 PRINT.P3M の設定例 (3)

```
1:  set PRINT.P3M=A:\etc\dvi2dsp.p3m
```

なお、環境変数 PRINT_P3M にも同じ効果がありますが、プリンタドライバは PRINT.P3M ⟶ PRINT_P3M の順に参照しますので、環境変数 PRINT.P3M の設定が優先されます[3)]。

3) いうまでもありませんが、プリンタドライバ実行時に、オプション -pk3fontman が指定された場合には、オプションの設定が優先されます。

1.5 ………………………… METAFONT の環境変数

METAFONT は以下の環境変数を参照します。

- MFBASES　base ファイルの検索パスの指定
- MFINPUTS　入力ファイルの検索パスの指定
- MFPOOL　mf.pool の検索パスの指定

本書の添付ディスクのインストーラでインストールした場合、これらの環境変数は適切に設定されていますから、METAFONT の動作について十分な知識がないうちにこれらの環境変数を変更することは避けてください。トラブルの原因になります。

なお、METAFONT は "\" ("¥") をパスの区切りとしては解釈しませんので、ディレクトリを指定するためには "\" ("¥") のかわりに "/" を用いなければなりません。

MFBASES　base ファイルの検索パスの指定

解説：　ファイル名の先頭に“&”を付加することでコマンドラインから指定された base ファイル (第2.5節「METAFONT のコマンドライン」(p.94) で詳説) を検索するディレクトリを指定します。“;”で区切ることで複数のパスを指定することができます。

```
A>set MFBASES=A:/MYBASE;B:/MBASE1/LOCALBASE
```

MFINPUTS　入力ファイルの検索パスの指定

解説：　コマンドラインやコントロール・シーケンス“\input”で指定されたファイルを検索するディレクトリを指定します。“;”で区切ることで複数のパスを指定することができます。

```
A>set MFINPUTS=A:/MYMFINPUT;B:/MINPUT/LOCALINPUT
```

MFPOOL　mf.pool の検索パスの指定

解説： METAFONT の実行ファイルが必要とする mf.pool を検索するディレクトリを指定します。

mf.pool は、METAFONT の実行ファイルがエラーメッセージやエラーに関する詳細な情報を出力する際に必要とするファイルで、inimf.x によって base ファイルを作成する時点で base ファイル内に読み込まれてしまいます。このため、環境変数 MFPOOL の設定は inimf.x の実行時にのみ必要とされますので、base ファイルを頻繁につくらない方は必要なときだけ指定しても実用上問題ありません。

```
A>set MFPOOL=A:/POOLDIR
```

1.6 ……………………………… makefont の環境変数

makefont.x が使用する環境変数には、

- MFHOME　METAFONT の HOME ディレクトリ
- MFBASES　base ファイルの収められているディレクトリ
- MFINPUTS　METAFONT ソースファイルの収められているディレクトリ

があります。

これらの環境変数が設定されていないときには、本書のインストーラで作成される環境で動作します。環境の変更、追加をした場合、これらの環境変数を指定してください。

MFHOME　METAFONT の HOME ディレクトリ

解説： makefont.x が参照するファイルは、基本的にこの環境変数が示すディレクトリ配下におきます。具体的にいえば、環境変数 MFHOME には、METAFONT システムのルートディレクトリを指定します。たとえば、本書の添付ディスク付属のインストーラで「一括インストール」[1]を行った場合、環境変数 MFHOME には %TEXHOME%/mf が設定されることになります。

1) 本書『Vol.1 — User's Guide 編』の第 2.2.3 項「一括インストール」(p.32) 参照。

MFBASES　METAFONT の base ファイルのディレクトリ

解説： METAFONT が参照するファイル cmplain.base, local.mf の存在するディレクトリを指定します。この環境変数が指定されていない場合は、環境変数 MFHOME で指定されているディレクトリ下の bases ディレクトリからファイルを探します。本書のインストーラでインストールした環境なら、この環境変数の指定は不要です。

MFINPUTS　METAFONT ソースファイルのディレクトリ

解説： makefont.x は、この環境変数が設定されていない場合、環境変数 MFHOME で示されるディレクトリ配下にある、以下に示すディレクトリから METAFONT ソースファイルを探しますが、この環境変数が指定されている場合、そのディレクトリからソースファイルを探すようになります。本書のインストーラでインストールした環境なら、この環境変数の指定は不要です。

- cmfonts
- LaTeXfonts
- utilityfonts\bases
- utilityfonts\black
- utilityfonts\gray
- utilityfonts\half
- utilityfonts\logo
- utilityfonts\manualfonts
- utilityfonts\mfbook
- utilityfonts\slant
- otherfonts

CHAPTER 2

Options

本章では、TEX、フォントマネージャ、プレビューア、プリンタドライバ、METAFONT、makefont という、それぞれの実行環境ごとに、実行ファイルのコマンドラインから入力した文字列引数の解釈について、あるいは、コマンドラインから指定することができる実行ファイルのオプション引数について説明します。

2.1 ································ TeX のコマンドライン

TeX の基本的な実行形式である initex.x と virtex.x のコマンドライン解釈について簡単に説明しておきます。

initex.x と virtex.x には通常のプログラムでコマンドラインから指定するオプションは存在しません。コマンドライン文字列は、そのまま TeX への入力文字として扱われます。

たとえば、コマンドラインから次のように入力したとします。

```
A> initex test.tex
```

この入力は、一見すれば test.tex をコマンドラインの引数として渡したように見えますが、これはただ単に、

```
A> initex
This is pTeX, C Version 2.99 j1.7 p1.0.9F EW (INITEX)
**test.tex
...
```

としたのとまったく等価になります。コマンドラインからの入力は、そのまま TeX の入力として扱われる点に注意してください。

TeX では、“&” を先頭に持つ最初のコマンドライン引数のみ、特別の意味を持ちます。“&” のあとに続く文字列には、initex.x で作成した拡張子 .fmt のファイルを指定します。詳しくは、『Vol.1 — User's Guide 編』の第 1.2.3 項「TeX の種類と実行ファイル」(p.10) を参照してください。

なお、再三述べたように、TeX は “\” (“¥”) をパスの区切りとは解釈しませんので、コマンドラインからディレクトリ付きのファイルネームを渡す場合は “\” (“¥”) のかわりに “/” を使用してください。

2.2 ……………………… FONTMAN オプション

2.2.1 FONTMAN オプション概説

フォントマネージャには、動作を指示するためのサブコマンドとしてのオプションが 6 つ[1]と、デバッグ情報を表示するためのオプションが 2 つあります。

サブコマンドとしてのオプションには、以下のものがあります。

- なし　通常登録
- -l　すべての登録フォントエイリアス・ネームの表示
- -a　すべての登録フォントエイリアスの表示
- -n　すべての登録フォントネームの表示
- -i　登録フォントドライバの情報表示
- -r　フォントドライバを 1 レベル解除する

“-i” に関しては、指定する引数によって動作が若干異なります。

フォントドライバの登録に失敗した場合には、デバッグ情報を表示しながら登録を行えば、どこが間違っているのかを知る手がかりになります。そのために 2 つの追加オプションが用意されています。

- -v　コンフィギュレーションファイルの行内容を逐次表示
- -V　-v の表示に加えてマクロ展開後の行内容も逐次表示

1)オプションを指定しない場合も含む。

2.2.2 FONTMAN オプション詳説

以下、各オプションについて、詳しく説明します。

引数なし　**フォントマネージャ常駐**

書式： fontman

機能： フォントマネージャを常駐させます。

解説： フォントマネージャを常駐させるだけです。
通常は、次に述べるように、コンフィギュレーションファイルを指定することによってフォントの登録も同時に行いますから、この方法によって起動することは、まずないでしょう。

〈コンフィギュレーションファイル名〉　**フォントマネージャ常駐 & フォント登録**

書式： fontman 〈コンフィギュレーションファイル名〉

機能： フォントマネージャを常駐させ、フォントを登録します。

解説： 第3.1節「コンフィギュレーションファイル文法」(p.100) で説明するコンフィギュレーションファイルを指定してフォントマネージャを常駐させ、コンフィギュレーションファイルの指示内容に従ってフォントを登録していきます。フォントマネージャの基本的な登録方法です。
付加オプションとして、“-v” か “-V” を併用できます (後述)。
わかりやすいように、コンフィギュレーションファイルの拡張子は .fm に統一してください。たとえば、mincho.fm などとします。添付ディスクのコンフィギュレーションファイルは myfonts.fm という名前ですから、これを使ってフォントマネージャを常駐させるには、次のようにします。

```
A>fontman myfonts.fm
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    フォントマネージャが常駐しました。
    コンフィギュレーションを読み込みます。
    ドライバを登録します。
    ジェネレータ : A:/usr/local/TeX/fontman/jxl4.sys
 -g -o 0 -p A:/usr/local/TeX/jxl4
        常駐しました。(font ID = $0001)
    ジェネレータ : A:/usr/local/TeX/fontman/jxl4.sys
 -m -o 0 -p A:/usr/local/TeX/jxl4
        常駐しました。(font ID = $0002)
    フィルタ : A:/usr/local/TeX/fontman/smooth.sys
 -s 256 -p 16
        常駐しました。(font ID = $0003 , 親 font ID = $0000)
    フィルタ : A:/usr/local/TeX/fontman/totex.sys
 -j A:/usr/local/TeX/fonts/goth10.tfm
 -e A:/usr/local/TeX/dump
        常駐しました。(font ID = $0004 , 親 font ID = $0003)
    フィルタ : A:/usr/local/TeX/fontman/smooth.sys
 -s 256 -p 24
        常駐しました。(font ID = $0005 , 親 font ID = $0000)
    フィルタ : A:/usr/local/TeX/fontman/totex.sys
 -j A:/usr/local/TeX/fonts/min10.tfm
 -e A:/usr/local/TeX/dump
        常駐しました。(font ID = $0006 , 親 font ID = $0005)
    登録は正常終了しました。
A>
```

なお、この実行例は、アスキーの『パーソナル日本語 TEX フォントライブラリ』の JXL4 フォーマットフォントを使用した場合のものです。他のフォントを使用している方は表示内容が異なります。
フォントドライバの登録は何回でも実行することができますが、2 回目以降、フォントマネージャの常駐自体を重ねて行うことはしません。この場合、フォントの追加登録だけを行います。その際、同じ名前のフォントがあった場合には上書きされます。なお、この方法でのフォントの追加登録は 32 回まで行えます。

-v 〈コンフィギュレーションファイル名〉

フォントマネージャ常駐 & フォント登録

書式： fontman -v 〈コンフィギュレーションファイル名〉

機能： フォントマネージャを常駐させ、フォントを登録します。

解説： 第3.1節「コンフィギュレーションファイル文法」(p.100) で説明するコンフィギュレーションファイルを指定してフォントマネージャを常駐させ、コンフィギュレーションファイルの指示内容に従ってフォントを登録していきます。

"-v" オプションを指定した場合は、読み込んだコンフィギュレーションファイルの行内容を逐次表示します。

p.27 で使用したコンフィギュレーションファイルを用いて実行例を挙げてみましょう。

```
A>fontman -v myfonts.fm
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by E x t (T.Kawamoto)
    フォントマネージャが常駐しました。
    コンフィギュレーションを読み込みます。
define driver = $env[TEXHOME]/fontman
define maxFontSize = 256
define jxl4 = $env[TEXHOME]/jxl4
                    :
                    :
font -tex-高速-明朝 = -通常|TotexMin
font -tex-高速-ゴシック = -強調|TotexGoth
font -tex-明朝 = $JXL4MIN
font -tex-ゴシック = $JXL4GOTH
    ドライバを登録します。
    ジェネレータ : A:/usr/local/TeX/fontman/jxl4.sys
 -g -o 0 -p A:/usr/local/TeX/jxl4
        常駐しました。(font ID = $0001)
    ジェネレータ : A:/usr/local/TeX/fontman/jxl4.sys
 -m -o 0 -p A:/usr/local/TeX/jxl4
                    :
                    :
```

このようにコンフィギュレーションファイルの内容が付加されて表示されます。フォントマネージャの常駐時、コンフィギュレーションファイルの不備が指摘された場合 (登録に失敗した旨が表示されます) には、この表示を参考にして対処することができます。

-V 〈コンフィギュレーションファイル名〉

フォントマネージャ常駐 & フォント登録

書式： fontman -V 〈コンフィギュレーションファイル名〉

機能： フォントマネージャを常駐させ、フォントを登録します。

解説： 第3.1節「コンフィギュレーションファイル文法」(p.100) で説明するコンフィギュレーションファイルを指定してフォントマネージャを常駐させ、コンフィギュレーションファイルの指示内容に従ってフォントを登録していきます。

```
A>fontman -V myfonts.fm
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    フォントマネージャが常駐しました。
    コンフィギュレーションを読み込みます。
define driver = $env[TEXHOME]/fontman
  ->    define driver = A:/usr/local/TeX/fontman
define maxFontSize = 256
  ->    define maxFontSize = 256
define jxl4 = $env[TEXHOME]/jxl4
  ->    define jxl4 = A:/usr/local/TeX/jxl4
define fonts = $env[TEXHOME]/fonts
  ->    define fonts = A:/usr/local/TeX/fonts
define dump = $env[TEXHOME]/dump
  ->    define dump = A:/usr/local/TeX/dump
define JXL4options = -o 0
  ->    define JXL4options = -o 0
dfilter MinSmooth = $driver/smooth.sys
 -s $maxFontSize -p 24
  ->    dfilter MinSmooth = A:/usr/local/TeX/fontman
/smooth.sys -s 256 -p 24
dfilter GothSmooth = $driver/smooth.sys -s $maxFontSize
 -p 16
  ->    dfilter GothSmooth = A:/usr/local/TeX/fontman
/smooth.sys -s 256 -p 16
generator JXL4Mincho = $driver/jxl4.sys -m $JXL4options
 -p $jxl4
  ->    generator JXL4Mincho = A:/usr/local/TeX/fontman
/jxl4.sys -m -o 0 -p A:/usr/local/TeX/jxl4
generator JXL4Gothic = $driver/jxl4.sys -g $JXL4options
 -p $jxl4
  ->    generator JXL4Gothic = A:/usr/local/TeX/fontman
/jxl4.sys -g -o 0 -p A:/usr/local/TeX/jxl4
                         :
```

“-V” オプションを指定した場合は、読み込んだ行の内容と、マクロ展開後の結果を逐次表示します。

p.27 で使用したコンフィギュレーションファイルを用いて前ページに実行例を挙げてみました。-> のあと、“-v” オプションを指定した場合に表示される情報以外に、コンフィギュレーションファイルの各行をマクロ展開した結果が表示されることに注目してください。“-v” オプションでコンフィギュレーションファイルの行内容を表示してもエラーに対処しきれない場合は、“-V” オプションを指定することにより、より詳細な情報を得ることができます。

-l　　フォント一覧

書式： `fontman -l`

機能： 登録フォントの一覧を表示します。

解説： すべての登録フォントのフォントエイリアスおよびフォントネームを表示します。フォントエイリアスやフォントネームについては、第3.1節「コンフィギュレーションファイル文法」(p.100)を参照してください。p.27 から使用している JXL4 フォーマットフォントを登録するコンフィギュレーションファイルを用いた場合の“-l”オプションによる実行例を挙げます。

```
A>fontman -l
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    -tex-高速-明朝
    -通常
    -tex-高速-ゴシック
    -強調
    -tex-明朝
    -tex-ゴシック
    sharp-rom
    大日本-ゴシック
    大日本-明朝
    sharp-rom-平滑(16)
    sharp-rom-平滑(16)-tex[goth]
    sharp-rom-平滑
    sharp-rom-平滑-tex[min]
A>
```

著作権表示の行より下にある名前のうち、ハイフン“-”から始まっているものがフォントエイリアスです。それ以外 (“sharp-” あるいは“大日本-”から始まっているもの) がフォントネームです。

-a　フォントエイリアス一覧

書式： fontman -a

機能： 登録フォントエイリアスの一覧を表示します。

解説： すべての登録フォントのフォントエイリアスについて、何の別名であるかを一覧表示します。

p.27 から使用している JXL4 フォーマットフォントを登録するコンフィギュレーションファイルを用いた場合の“-a”オプションによる実行例を挙げます。p.31 の“-l”オプションによってフォントエイリアスとフォントネームを表示した場合と比較してください。

```
A>fontman -a
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    -tex-高速-明朝 = sharp-rom-平滑-tex[min]
    -通常 = sharp-rom-平滑
    -tex-高速-ゴシック = sharp-rom-平滑(16)-tex[goth]
    -強調 = sharp-rom-平滑(16)
    -tex-明朝 = 大日本-明朝
    -tex-ゴシック = 大日本-ゴシック
A>
```

-n　　フォントネーム一覧

書式： fontman -n

機能： フォントネームの一覧を表示します。

解説： すべての登録フォントのフォントネームについて、一覧を表示します。p.27 から使用している JXL4 フォーマットフォントを登録するコンフィギュレーションファイルを用いた場合の "-n" オプションによる実行例を挙げます。p.31 の "-l" オプションによってフォントエイリアスとフォントネームを表示した場合と比較してください。

```
A>fontman -n
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    sharp-rom
    大日本-ゴシック
    大日本-明朝
    sharp-rom-平滑(16)
    sharp-rom-平滑(16)-tex[goth]
    sharp-rom-平滑
    sharp-rom-平滑-tex[min]
A>
```

"-l" オプションを使用した場合の表示から、"-" で始まるエントリ (フォントエイリアス) を除いたものと同じになります。

-i　　登録フォントドライバ情報一覧

書式：　fontman -i

機能：　すべての登録フォントドライバの情報を表示します。

解説：　すべての登録フォントドライバについて、詳細情報 (登録時のドライバオプション) を表示します。

```
A>fontman -i
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    sharp-rom
        ＲＯＭフォント：depth = $0000 , font ID = $0000
    大日本-ゴシック
        ビットマップジェネレータ：  depth = $0001 ,
 font ID = $0001
        A:/usr/local/TeX/fontman/jxl4.sys -g -o 0
 -p A:/usr/local/TeX/jxl4
    大日本-明朝
        ビットマップジェネレータ：  depth = $0001 ,
 font ID = $0002
        A:/usr/local/TeX/fontman/jxl4.sys -m -o 0
 -p A:/usr/local/TeX/jxl4
    sharp-rom-平滑(16)
        ビットマップフィルタ：      depth = $0001 ,
 font ID = $0003 , 親 font ID = $0000
        A:/usr/local/TeX/fontman/smooth.sys -s 256 -p 16
    sharp-rom-平滑(16)-tex[goth]
        ビットマップフィルタ：      depth = $0001 ,
 font ID = $0004 , 親 font ID = $0003
        A:/usr/local/TeX/fontman/totex.sys
 -j A:/usr/local/TeX/fonts/goth10.tfm
 -e A:/usr/local/TeX/dump
    sharp-rom-平滑
        ビットマップフィルタ：      depth = $0001 ,
 font ID = $0005 , 親 font ID = $0000
        A:/usr/local/TeX/fontman/smooth.sys -s 256 -p 24
    sharp-rom-平滑-tex[min]
        ビットマップフィルタ：      depth = $0001 ,
 font ID = $0006 , 親 font ID = $0005
        A:/usr/local/TeX/fontman/totex.sys
 -j A:/usr/local/TeX/fonts/min10.tfm
 -e A:/usr/local/TeX/dump
A>
```

p.27 から使用している JXL4 フォーマットフォントを登録するコンフィギュレーションファイルを用いた場合の “-i” オプションによる実行例を前ページに挙げておきました。

-i=$ 〈数値〉　指定フォントドライバ情報

書式：　fontman -i=$ 〈数値〉

機能：　指定フォントドライバの情報を表示します。

解説：　〈数値〉で指定したフォントドライバについて、詳細情報 (登録時のドライバオプション) を表示します。なお、〈数値〉は 16 進数で指定します。

p.27 から使用している JXL4 フォーマットフォントを登録するコンフィギュレーションファイルを用いた場合、“font ID = $0003” 等の部分がフォントドライバの ID です。

フォントドライバには、登録された順番に「フォント ID」と呼ばれる情報が付与されます。そして、各フォントドライバはこの ID をもとにしてフォントデータのやりとり (入出力) を行います。つまり、あるフォントドライバについて注目してみると、その親 ID を持つフォントドライバの出力は当該フォントドライバの入力に、自身の ID は当該フォントドライバの出力にあたります。したがって、フォント ID を追っていけば、そのフォントがどのフォントドライバによって処理されたものかわかります[2)]。

このフォント ID を本オプションで指定すると、そのフォントドライバだけの情報表示に限定されます。たとえば、“font ID=$0003” についての表示を行いたいなら、以下のようにします。

```
A>fontman -i=$0003
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    sharp-rom-平滑 (16)
        ビットマップフィルタ：      depth = $0001 ,
 font ID = $0003 , 親 font ID = $0000
        A:/usr/local/TeX/fontman/smooth.sys -s 256 -p 16
A>
```

2)FONTMAN Ver.1 では、このフォント ID をもとにコンフィギュレーションファイルを作成しなければならなかったため、コンフィギュレーションファイルの作成は必ずしも簡単なものではありませんでした。

-i= 〈フォントネーム〉　指定フォントドライバ情報

書式：　fontman -i= 〈フォントネーム〉

機能：　指定フォントドライバの情報を表示します。

解説：　〈フォントネーム〉で指定したフォントドライバについて、詳細情報(登録時のドライバオプション) を表示します。なお、〈フォントネーム〉にはフォントエイリアスも指定可能です。

p.27 から使用している JXL4 フォーマットフォントを登録するコンフィギュレーションファイルを用いた場合、大日本-明朝 フォントについて表示を行いたいなら、以下のようにします。

```
A>fontman -i=大日本-明朝
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    大日本-明朝
        ビットマップジェネレータ：  depth = $0001 ,
 font ID = $0002
        A:/usr/local/TeX/fontman/jxl4.sys -m -o 0
 -p A:/usr/local/TeX/jxl4
A>
```

-r　　フォントドライバ解除

書式：　fontman -r

機能：　フォントドライバを解除します。

解説：　フォントドライバを1レベル (1depth) 解除します。そのレベルで使用していたメモリはこの時点で解放されます。すべてのドライバが解除されたら、フォントマネージャ自体も解除されます。

たとえば、二重にフォントマネージャを登録していた場合、1度目の解除では、2回目の常駐時に登録されたフォントドライバだけが解除されます。

```
A>fontman -r
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    1 depth 分のフォントドライバが解除されました。
A>
```

2度目の解除で、1回目の常駐時に登録されたフォントドライバ、およびフォントマネージャが解除されます。

```
A>fontman -r
X68k Font Manager Ver 3.00c Copyright (C) 1989,90,91,92
 by Ｅｘｔ (T.Kawamoto)
    1 depth 分のフォントドライバが解除されました。
    フォントマネージャは解除されました。
A>
```

通常使用する場合は、「フォントマネージャは解除されました。」というメッセージが出るまで解除操作を繰り返してください。

2.3 .... PREVIEW オプション

2.3.1 PREVIEW オプション概説

プレビューアには、以下のオプションがあります。

- `-remark` 注釈文
- `-info` 情報表示
- `-dpi` 解像度指定
- `-mag` 拡大率指定
- `-timer` キースキャン・タイマ・インターバル指定
- `-highReso` 高解像度指定
- `-GRAM` グラフィック RAM 使用宣言
- `-buf` ページバッファ数指定
- `-width` バッファ横ドット数指定
- `-height` バッファ縦ドット数指定
- `-xOffset` 左上座標指定 (X 座標)
- `-yOffset` 左上座標指定 (Y 座標)
- `-paperCol` 背景色の指定
- `-penCol` ペンの色の指定
- `-pk2fontman` FONTMAN Ver.2 の変換ファイル指定
- `-pk3fontman` FONTMAN Ver.3 の変換ファイル指定
- `-fontmanMaxSize` フォントマネージャの最大フォントサイズ
- `-layout` レイアウト表示指定
- `-subst` 代用表示方法指定

なお、これらは大文字、小文字の区別をしません。

2.3.2 PREVIEW オプション詳説

以下、各オプションについて、詳しく説明します。

-remark　注釈文

書式： -remark=[string]

機能： 注釈文を書きます。

解説： [string] には、半角スペースを含めないでください。かわりに、かなスペースまたは全角スペースを使用します。プレビューアのコンフィギュレーションファイルにコメントを残したい場合、これを利用します。

List 2-1● -remark を使用した例

```
1:  -remark=　preview.cfg　for　Ｅｘｔ
2:  -buf=9 -remark=バッファは前後それぞれ4画面
3:  -GRAM　-remark=グラフィックRAMは未使用でもったいないから使う
```

List 2-1 の 1 行目は、コメント行として使用する例です。2 行目や 3 行目では、各行の右側にコメントを併記する形式の例です。

[string] のなかに使用するスペースとしては、かなスペースはわかりにくいので、かわりにアンダースコア“_”を使うのも手です。

-info 情報表示

書式： -info

機能： 各種情報を表示して終了します。

解説： テキストの縦横のドット数や総ページ数等、各種情報を表示し、そのまま終了します。

実行例を挙げます。

```
A>preview -info preview
X680x0 TeX Previewer Ver 2p09a Copyright 1989,90,
91,92,93 by Ｅｘｔ (T.Kawamoto)
       Tpic Specials support Ver 1.00a Copyright
 1993 by TSG+SHIMA
font manager     ：version 3
フリースペース  ：  3242  K bytes
  内ＧＲＡＭ     ：  510  K bytes
総バッファ数     ：  7
バッファ幅、高さ：   1024  ,   2048
テキスト左上座標：(   118  ,    118)
テキスト右下座標：(   855  ,   1260)
フォーマット     ：ＴｅＸ
基本ＤＰＩ       ：118
拡大率           ：1
総ページ数       ：11
フォント         ：
  5: 118/cmr6.pk -> A:/usr/local/TeX/pks/118/cmr6.pk
  7: 118/cmsy6.pk -> dummy font
 10: 118/cmr7.pk -> A:/usr/local/TeX/pks/118/cmr7.pk
 16: 118/cmr8.pk -> A:/usr/local/TeX/pks/118/cmr8.pk
 18: 118/cmsy8.pk -> A:/usr/local/TeX/pks/118/cmsy8.pk
 21: 118/min8.pk -> -tex-高速-明朝 (  13.0621)
                        ：
                        ：
```

○ 「font manager」の行は、フォントマネージャのバージョンを表示しています。

○ 「フリースペース」の行は、グラフィック RAM も含めて、プレビューアがワーキングエリア[1]として使用可能な空きエリアの容量を、Kバイト単位で表示しています。

○ 「内 GRAM」の行は、フリースペースのうち、グラフィック RAM に割り当てられた部分をKバイト単位で表示しています。

○ 「総バッファ数」の行は、プレビューアが確保しているページバッファの個数です。この個数の指定には、オプション“-buf”を使用してください。

1) マニュアル類には、cell 領域と述べられている部分。

○ 「バッファ幅、高さ」の行から「テキスト右下座標」の行までは、ページバッファのサイズを表示しています。座標は、バッファの左上隅を原点にとり、右方向および下方向が正として表示されます。

○ 「フォーマット」の行は、dvi ファイルが通常の横書き TEX のものか、縦書き TEX のものであるかを示しています。

○ 「基本 DPI」の行は、オプション“-dpi”で指定した値であって、実際の画面の dpi 値ではありません。実際の画面の dpi 値は約 71dpi ですから、この行の表示が 71 より大きければ大きいほど、画面上にはより拡大されて表示されることになります。

○ 「拡大率」の行は、オプション“-mag”で指定した率を表示しています。

○ 「総ページ数」の行は、dvi ファイルに含まれているページの総数です。

○ 「フォント」の行から下の部分は、dvi ファイルで使用されているフォントの一覧です。ここをよく見ると、代用フォントの状態がわかります。詳しくはオプション“-subst”の項を参照してください。

-dpi　解像度指定

書式： -dpi=[num]

機能： 解像度を直接指定します。

解説： 単位は dpi です。本オプションを指定していないとき、プレビューアの解像度には 118dpi が採用されます。この値は、16 ドットフォントの漢字がほぼ 10 ポイントになるように決められました。

ただし、実際のディスプレイの解像度は、実測によると約 71dpi です。したがって、71dpi 用のフォントを別途作成し、“preview -dpi=71” を実行すれば、画面上に正確なサイズで表示することができます。

以下に示すのが、実測のために使用した「物差し」です。

List 2-2● 定規のサンプル — ruler.tex

```
 1: \documentstyle{jreport}
 2: \begin{document}
 3: \begin{center}
 4: \unitlength 1mm
 5: \newcounter{num}
 6: \begin{picture}(100,10)
 7: \put(0,10){\line(1,0){100}}
 8: \multiput(0,10)(1,0){101}{\line(0,-1){2}}
 9: \multiput(5,10)(10,0){10}{\line(0,-1){3}}
10: \multiput(0,10)(10,0){11}{\line(0,-1){5}}
11: \end{picture}
12: \end{center}
13: \thispagestyle{empty}
14: \end{document}
```

このファイル ruler.tex を LaTeX にかけ、プレビューアで見てみましょう (ruler.tex は、71dpi のフォントを作成しなくても見ることができるように、目盛等の数字表示は省略しています)。

```
A>latex ruler
A>preview -dpi=71 ruler
```

画面に本物の物差しを当てて測ってみてください。正しいスケールであることが確認できるでしょう。もちろん、この「物差し」はプリンタに出力することもできます。

```
A>print ruler
```

-mag　拡大率指定

書式 1：　-mag=[num]

書式 2：　-mag=half

機能：　拡大率を指定します。

解説：　この指定に従って、表示拡大率が変化します。このオプションの指定がない場合、dvi ファイルに書き込まれている値が採用されます。各指定値に対応する実際の拡大率とフォントの dpi 値を以下に挙げます。「フォント」の項目は、“-dpi=118”と指定した場合、拡大後、どのフォントが主に使われるかを示したものです。

指定	拡大率	フォント
0	1.000 倍	118dpi
half	1.096 倍	129dpi
1	1.200 倍	142dpi
2	1.440 倍	170dpi
3	1.728 倍	204dpi
4	2.074 倍	245dpi
5	2.488 倍	294dpi

-timer　キースキャン・タイマ・インターバル指定

書式：　-timer=[num]

機能：　キースキャンのためのタイマ・インターバルを指定します。

解説：　数値が小さいほどスクロールが速くなります。あまり数値が小さいと書き込みが行われなくなる恐れがあるので、お勧めできません。デフォルトは 60 です。

次に、少しキースキャン・タイマ・インターバルを速くしてみる場合の指定例を挙げます。

```
A>preview -timer=50 sample
```

-highReso　高解像度指定

書式：　-highReso

機能：　1024 × 848 ドットの表示モードにします。

解説：　このとき、さらにテキスト RAM の一部を cell 領域[2] に使用します。

```
A>preview -highReso sample
```

1024 × 848 ドットの表示モードでは画面がちらつきますが、これはハードウェアの制限からくる症状で、デバイスドライバのトラブルではありません。

2) TEX ドライバが使うワーキングエリア。

-GRAM　グラフィック RAM 使用宣言

書式：　-GRAM

機能：　グラフィック RAM も cell 領域に使用します。

解説：　グラフィック RAM も cell 領域に使用したい場合に指定します。
グラフィック RAM が他の目的に使用されていた場合はエラーを表示して終了します。

```
A>preview -GRAM sample
X680x0 TeX Previewer Ver 2p09a Copyright 1989,90,91,92,93
 by Ｅｘｔ (T.Kawamoto)
       Tpic Specials support Ver 1.00a Copyright 1993
 by TSG+SHIMA

Program Stop : graphic ＲＡＭは使用できません。
```

この表示が出た場合には、グラフィック RAM を専有しているアプリケーション[3] を外し、使用できるようにしてください。
もし、そのアプリケーションを外したくなければ、プレビューアでのグラフィック RAM の使用 (-GRAM) をあきらめるしかありません。

3) 最も可能性の高いものは、グラフィック RAM ディスクでしょう。

-buf　ページバッファ数指定

書式：　-buf=[num]

機能：　ページバッファを何ページ分メモリにとるかを指定します。

解説：　このオプションを省略した場合、デフォルトとして 2 が採用されます。現在のページを中心にして、どのページをメモリにためるかを指定します。現在のページが 5 ページのときの例をバッファ数別に挙げておきます。

バッファ数	ためるページ
2	5 6
3	4 5 6
4	4 5 6 7
5	3 4 5 6 7
6	3 4 5 6 7 8
7	2 3 4 5 6 7 8

前後のページにすぐに移りたいのなら、最低でも 4 ページ分のバッファがないと快適な環境にはならないでしょう。

-width　バッファ横ドット数指定

書式：　-width=[num]

機能：　バッファの横幅をドット単位で指定します。

解説：　ただし、p.45 で述べたオプション "**-highReso**" が指定されていないと、本オプションの指定は意味がありません。デフォルトは 1024 です。また、指定する値は 16 の倍数でなければなりません。

なお、表示対象の dvi ファイルの横幅を超えて指定することはできません。

使用例を挙げます。

```
A>preview -highReso -width=1536 sample
```

プレビューアで閲覧可能な幅が 1.5 倍に増えます。

-height　バッファ縦ドット数指定

書式：　-height=[num]

機能：　バッファの縦幅をドット単位で指定します。

解説：　ただし、p.45 で述べたオプション “-highReso” が指定されていないと、本オプションの指定は意味がありません。デフォルトは 2048 です。また、指定する値は 16 の倍数でなければなりません。

なお、表示対象の dvi ファイルの縦幅を超えて指定することはできません。

使用例を挙げます。

```
A>preview -highReso -height=3072 sample
```

プレビューアで閲覧可能な高さが 1.5 倍に増えます。

-xOffset　左上座標指定（X 座標）

書式：　-xOffset=[num]

機能：　画面に対して書き込む位置をドット単位で指定します。

解説：　次ページで説明する“-yOffset”と対にして使用します。

座標系は画面左上を原点とし、右方向が X 軸の正の方向、下方向が Y 軸の正の方向です。これらのスイッチで、どの位置から書き込みを始めるかを指定します。たとえば、“-xOffset=0”と指定します。

```
次ページで説明する -yOffset と対にして使用します。
座標系は画面左上を原点とし、右方向が X 軸の正の方向、下方向
が Y 軸の正の方向です。これらのスイッチでどの位置から書き込みを始
めるかを指定します。
```

これを、“-xOffset=100”と指定すると、下の例のように 100 ドット右方向にずれます。

```
          次ページで説明する -yOffset と対にして使用します。
          座標系は画面左上を原点とし、右方向が X 軸の正の方向、下方向
          が Y 軸の正の方向です。これらのスイッチでどの位置から書き込
          めるかを指定します。
```

負の値を指定すると、逆に左方向にずれます。デフォルトとしては、画面の中央に印刷用紙の中央がくるような値が採用されています。

-yOffset　左上座標指定 (Y 座標)

書式：　-yOffset=[num]

機能：　画面に対して書き込む位置をドット単位で指定します。

解説：　前ページで説明した“-xOffset”と対にして使用します。

座標系は、画面左上を原点とし、右方向が X 軸の正の方向、下方向が Y 軸の正の方向です。これらのスイッチでどの位置から書き込みを始めるかを指定します。たとえば、“-yOffset=0”と指定します。

```
前ページで説明した -xOffset と対にして使用します。
座標系は画面左上を原点とし、右方向が X 軸の正の方向、下方向
が Y 軸の正の方向です。これらのスイッチでどの位置から書き込みを始
めるかを指定します。
```

これを、“-yOffset=100”と指定すると、下の例のように 100 ドット下方向にずれます。

```



前ページで説明した -xOffset と対にして使用します。
座標系は画面左上を原点とし、右方向が X 軸の正の方向、下方向
```

負の値を指定すると、逆に上方向にずれます。デフォルトとしては、画面の中央に印刷用紙の中央がくるような値が採用されています。

-paperCol　背景色の指定

書式： -paperCol=[num]

機能： ディスプレイ画面の背景色の指定です。

解説： [num] はカラーコードで、デフォルトは $C630（白）です。十進数で指定します。

カラーは、緑成分・赤成分・青成分ごとに 0 から 31 までの数値で指定することができます。緑成分を 2048 倍したものと、赤成分を 64 倍したものと、青成分を 2 倍したものを足したものが、カラーコードとなります。

たとえば、緑成分 30、赤成分 30、青成分 28 の場合は、$30 \times 2048 + 30 \times 64 + 28 \times 2 = 63416$ となります。

使用例を挙げます。ディスプレイ画面の背景色を、紙らしくなるように変えてみましょう。

```
A>preview -paperCol=63416 sample
```

-penCol　ペンの色の指定

書式： -penCol=[num]

機能： ペンの色の指定です。

解説： [num] はカラーコードで、デフォルトは $0000（黒）です。十進数で指定します。

カラーは、緑成分・赤成分・青成分ごとに 0 から 31 までの数値で指定することができます。緑成分を 2048 倍したものと、赤成分を 64 倍したものと、青成分を 2 倍したものを足したものが、カラーコードとなります。

たとえば、緑成分 0、赤成分 5、青成分 13 の場合は、$0 \times 2048 + 5 \times 64 + 13 \times 2 = 346$ となります。

この場合、ペンの色はほんの少し青味がかったものになります。

```
A>preview -penCol=346 sample
```

-pk2fontman　FONTMAN Ver.2 の変換ファイル指定

書式： -pk2fontman=[string]

機能： FONTMAN Ver.2 の変換ファイル指定です。

解説： FONTMAN Ver.2 を利用する場合、preview.p2m ファイルを必要とします。これは、デフォルトではカレントディレクトリの preview.p2m、あるいは、%TEXHOME%/preview.p2m を参照しますが、このオプションを指定することで変更することができます。また、環境変数 PREVIEW.P2M または PREVIEW_P2M でも設定可能です。環境変数とオプションの両方で指定した場合、オプションのほうが優先されます。ファイル名をフルパスで指定してください。詳しくは第1.3節「PREVIEW 環境変数」(p.7) を参照してください。
なお、本書には FONTMAN Ver.2 は添付されていません。

-pk3fontman　FONTMAN Ver.3 の変換ファイル指定

書式： -pk3fontman=[string]

機能： FONTMAN Ver.3 の変換ファイル指定です。

解説： FONTMAN Ver.3 を利用する場合、preview.p3m ファイルが必要になります。FONTMAN Ver.3 は、デフォルトではカレントディレクトリの preview.p3m、あるいは %TEXHOME% の preview.p3m を参照しますが、-pk3fontman オプションで変更することができます。
環境変数 PREVIEW.P3M または PREVIEW_P3M でも設定可能です。オプションと環境変数の両方で指定した場合は、オプションの指定のほうが優先されます。
ファイル名をフルパスで指定してください。詳しくは、第1.3節「PREVIEW 環境変数」(p.7) を参照してください。

-fontmanMaxSize フォントマネージャの最大フォントサイズ

書式： -fontmanMaxSize=[num]

機能： フォントマネージャの最大フォントサイズを設定します。

解説： FONTMAN Ver.2 あるいは Ver.3 を利用すると、漢字フォントをかなり大きなサイズまで利用することが可能になります。フォントマネージャとのインターフェースでは、フォントのバッファを必要なサイズだけ確保しなければならないので、このオプションで指定します。デフォルトでは 512 ドットで、普通はこれで十分です。

-layout レイアウト表示指定

書式： -layout

機能： レイアウト表示を指定します。

解説： すべてのフォントのすべての文字がボックスで表示されます。スピードが若干速くなりますが、たいした効果は期待できません。

-subst　代用表示方法指定

書式：　-subst=[string]

機能：　代用表示方法を指定します。

解説：　フォントファイルがない場合の動作を指定します。[string] は、box, cmrmin, nofont のうちから選んでください。それぞれの動作は以下のようになっています。

(1)　-subst=box

フォントファイルがない場合、ボックスを表示します。ただし、tfm ファイルがないとボックス表示はできません。その場合、ダミーフォントになります。

(2)　-subst=cmrmin

フォントファイルがない場合、cmr[4] か min[5] で代用します。この場合、フォントの領域を示すボックスは正確ではなくなります。なお、これらのフォントが見つからなかった場合、代用表示はできません。その場合、ダミーフォントとなります。
この指定で代用を行った場合、“-info” で見ると “check sum error” がよく出ますが、無視してもかまいません。

(3)　-subst=nofont

フォントファイルが見つからなかった場合、ダミーフォントとなります。

なお、上述のいかなる場合でもダミーフォントが見つかった場合は、途中で動作を打ち切ります。該当フォントを作成したうえでリトライしてください。

4)英語フォントの場合。

5)日本語フォントの場合。

2.4 ································ PRINT オプション

2.4.1 PRINT オプション概説

プリンタドライバのオプションは、大きく分けて次の 3 種類に分類できます。

(1) ドライバ制御系
(2) 汎用プリンタ制御系
(3) ラスタプリンタ制御系

それぞれに属するオプションの一覧を挙げます。なお、これらは大文字、小文字の区別をしません。

◆ ドライバ制御系

- `-remark` 注釈文
- `-info` 情報表示
- `-dpi` 解像度指定
- `-mag` 拡大率指定
- `-timer` キースキャン・タイマ・インターバル指定
- `-TRAM` テキスト RAM 使用宣言
- `-GRAM` グラフィック RAM 使用宣言
- `-landscape` 横置き (縦方向) 印字指定 (右回転)
- `-vertical` 横置き (縦方向) 印字指定 (右回転)[1]
- `-turnRight` 横置き (縦方向) 印字指定 (右回転)[2]
- `-turnLeft` 横置き (縦方向) 印字指定 (左回転)
- `-width` バッファ横ドット数指定
- `-height` バッファ縦ドット数指定
- `-xOffset` 左上座標指定 (X 座標)
- `-yOffset` 左上座標指定 (Y 座標)

1)-landscape と同じ。
2)-landscape と同じ。

- `-pk2fontman` FONTMAN Ver.2 の変換ファイル指定
- `-pk3fontman` FONTMAN Ver.3 の変換ファイル指定
- `-fontmanMaxSize` フォントマネージャの最大フォントサイズ
- `-layout` レイアウト印字指定
- `-subst` 代用印字方法指定
- `-dump` ダンプ出力

これらのオプションは、コマンドラインで直接指定するのが普通の使い方です。ただし、よく使うオプションはコンフィギュレーションファイルに入れておくのもいいでしょう。

◆ 汎用プリンタ制御系

- `-MSBisUpper` 印字ヘッド方向指定
- `-MSBisLeft` 印字ヘッド方向指定
- `-pinBytes` 印字ヘッドのピン数指定
- `-pinHeights` 印字ヘッドのピン数指定
- `-prBufSize` プリンタバッファの大きさ指定
- `-init` プリンタ初期化のコード列指定
- `-CRLF` プリンタ改行のコード列指定
- `-extraCRLF` 空行改行時の追加コード列指定
- `-FF` プリンタ改ページのコード列指定
- `-graphic` ドットグラフィック指定のコード列指定
- `-start` ドット単位印字開始位置指定のコード列指定
- `-relative` ドット単位印字位置相対指定のコード列指定
- `-repeat` グラフィックリピート指定のコード列指定

これらはプリンタを制御するオプション類ですから、通常はコンフィギュレーションファイル `print.cfg` に書いておく性質のものです。したがって、すでに TEX の HOME ディレクトリ内に `print.cfg` がある読者（インストール時にプリンタを指定して `print.cfg` ができあがっている読者）の場合は、これらのオプションはすでに設定済みです。あらためてコマンドラインから指定する必要はありません。

不幸にも該当するコンフィギュレーションファイルが見つからなかった人は、本節とプリンタのマニュアルをよく読んでからオプションを指定して、コンフィギュレーションファイルを作成しなければなりません。作成のしかたについては、『Vol.1 — User's Guide 編』の第 5.3 節「`print.cfg` の作り方」(p.186) を参照してください。

もし新しいコンフィギュレーションファイルが作成できましたら、お手数ですが、筆者（川本）宛に送ってください。筆者がパソコン通信を通じて他のユーザ

に広めるなり、この本の新版に収録させていただくなりしたいと思います。そうすれば、同様の悩みを抱えている他の読者の手助けになることと思いますので。

さて、このなかで -init, -CRLF, -extraCRLF, -FF, -graphic, -start, -relative, -repeat については、引数として文字列を指定します。指定する際はエスケープ・シーケンス等を使いますが、使用可能なコードについての詳細は第2.4.3項「-switch=[string] で使用できるコード」(p.91)を参照してください。

そのうち、-graphic, -start, -relative, -repeat は、グラフィックイメージをプリンタに送出する際に使用するコード列を指定するものです。4つありますが、どれも同じ目的を遂行するためにあります。プリンタによっては、どれかひとつしかサポートしていないことがあります。

◆ ラスタプリンタ制御系

- -Raster　　ラスタスキャン出力指定
- -MH　　MH 符号圧縮指定
- -EveryRaster　　出力ラスタ数指定
- -RasterOutPutOrder　　ラスタ情報出力順序指定
- -Raster_xPos　　ラスタスキャン x 方向位置のコード列指定
- -Raster_yPos　　ラスタスキャン y 方向位置のコード列指定
- -Raster_xSize　　ラスタスキャン x 方向サイズのコード列指定
- -Raster_ySize　　ラスタスキャン y 方向サイズのコード列指定

これらは、ラスタスキャン系のプリンタを使用する場合に使うオプションで、汎用プリンタ系のオプションと同じく、プリンタを制御するオプション類です。したがって、通常はコンフィギュレーションファイル print.cfg に書いておく性質のものです。ただし、-EveryRaster は dvi ファイルによってはコマンドラインで指定するのが効果的です。

2.4.2　PRINT オプション詳説

以下、各オプションについて、詳しく説明を行います。

-remark　注釈文

書式： -remark=[string]

機能： 注釈文を書きます。

解説： [string] には、半角スペースを含めないでください。かわりにかなスペースあるいは全角スペースを使用します。print.cfg に覚え書きを残したい場合、これを利用します。

List 2-3● -remark を使用した例

```
 1: -remark=ＣＺ－８ＰＣ３用 print.cfg by ＪｕｎＫ
 2: -dpi=180
 3: -MSBisUpper
 4: -pinBytes=3
 5: -init=\ec1\e%%9\x10\x20\x08
 6:           -remark=美しくないけどよしとしよう by じぱんぐ
 7: -CRLF=\r\n
 8: -FF=\f       -remark=corrected by じぱんぐ 89/12/11
 9: -graphic=\eJ%2m
10: -start=\e\x10%4d
11: -repeat=
```

[string] のなかに使用するスペースとしては、かなスペースだとわかりにくいので、そのかわりにアンダースコア“_”を使うのも手です。

-info 情報表示

書式： -info

機能： 各種情報を表示して終了します。

解説： テキストの縦横のドット数や総ページ数等、各種情報を表示し、そのまま終了します。

実行例を挙げます。

```
A> print -info preview
X680x0 TeX Printer Driver Ver 2p09a Copyright 1989,90,
91,92,93 by Ｅｘｔ (T.Kawamoto)
       Tpic Specials support Ver 1.00a Copyright 1993
 by TSG+SHIMA
       RasterOutput Devices support Ver 1.00 Copyright
 1992, 1993 by T. Hilano
font manager     ：version 3
フリースペース   ： 3903 K bytes
プリンタバッファ： 256 K bytes
プリンタピン数   ： 24 ピン
  印字絶対位置 リピート 指定。
バッファ幅、高さ：  1440 ,  1980
テキスト左上座標：(  157 ,   118)
テキスト右下座標：( 1282 ,  1861)
  注意： 左端が 23 dots 右端が 23 dots 上端が 62 dots 下端が
 62 dots 紙面から切れます。
フォーマット     ：ＴｅＸ
基本ＤＰＩ       ：180
拡大率           ：1
総ページ数       ：11
フォント         ：
  5: 180/cmr6.pk -> A:/usr/local/TeX/pks/180/cmr6.pk
  7: 180/cmsy6.pk -> A:/usr/local/TeX/pks/180/cmsy6.pk
 10: 180/cmr7.pk -> A:/usr/local/TeX/pks/180/cmr7.pk
 16: 180/cmr8.pk -> A:/usr/local/TeX/pks/180/cmr8.pk
 18: 180/cmsy8.pk -> A:/usr/local/TeX/pks/180/cmsy8.pk
 21: 180/min8.pk -> -tex-明朝 (  19.9253)
  ....
```

○ 「font manager」の行は、フォントマネージャのバージョンを表示しています。

○ 「フリースペース」の行は、プリンタドライバがワーキングエリア[3]として使用可能な空きエリアの容量をＫバイト単位で表示しています。

○ 「プリンタバッファ」の行は、プリンタスプーリング用に確保されたバッファの容量をＫバイト単位で表示しています。

3)マニュアル類には、cell 領域と述べられている部分。

○ 「プリンタピン数」の行は、プリンタの印字ピンの数を、8 本をひとまとまりとして表示しています。

○ 「プリンタピン数」の次の行には、プリンタ関連のその他の情報が表示されています。ここに表示される情報は、以下のようなものです。

右回転, 左回転　用紙を 90 度回転して印字する場合に、右回転[4] するか左回転[5] するかの表示。

ラスタスキャン　ラスタスキャン[6]のプリンタであるかどうかの表示。

MH 符号圧縮　ラスタスキャンの場合に、MH 符号圧縮[7]を行うかどうかの表示。

ビット逆順　ビット並びが逆順[8]のプリンタを使用しているかどうかの表示。

印字絶対位置　印字絶対位置のシーケンス[9]を用いているかどうかの表示。

印字相対位置　印字相対位置のシーケンス[10]を用いているかどうかの表示。

リピート　グラフィックリピートのシーケンス[11]を用いているかどうかの指定。

それぞれ該当するオプションの説明の項を参照してください。

○ 「バッファ幅、高さ」の行から「テキスト右下座標」の行までは、ページバッファのサイズを表示しています。座標は、バッファの左上隅を原点に取り、右方向および下方向が正として表示されます。

○ 「テキスト座標」の次の行では、ページバッファが小さすぎるために出力する内容がバッファ領域 (紙面) をはみ出すときに注意を促します。ただし、上下左右 1 インチの余裕を見込んでいますので、はみ出した量が 1 インチを超えないかぎり大丈夫です。ドット数とインチサイズとの対応については、あなたのプリンタの dpi 値を参考にしてください。たとえば 360 dpi のプリンタであれば、1 インチが 360 ドット分となります。

○ 「フォーマット」の行は、dvi ファイルが通常の横書き TeX のものか、縦書き TeX のものであるかを示しています。

○ 「基本 DPI」の行は、オプション "-dpi" で指定した値です。あなたのプリンタの dpi 値と違っていれば、オプション "-dpi" で正しく設定しなければなりません。

○ 「拡大率」の行は、オプション "-mag" で指定した率を表示しています。

○ 「総ページ数」の行は、dvi ファイルに含まれているページの総数です。

4) -landscape 等同等オプション指定時。p.65参照。

5) -turnLeft 指定時。p.66 参照。

6) -Raster 指定時。p.86 参照。

7) -MH 指定時。p.87参照。

8) -MSBisUpper 指定時。p.74参照。

9) -start 指定時。p.83参照。

10) -relative 指定時。p.84 参照。

11) -repeat 指定時。p.85 参照。

○ 「フォント」の行から下の部分は、dvi ファイルで使用されているフォントの一覧です。ここをよく見ると、代用フォントの状態がわかります。詳しくは、オプション “-subst” の項 (p.72) を参照してください。

-dpi　解像度指定

書式：　-dpi=[num]

機能：　解像度を直接指定します。

解説：　単位は dpi です。本オプションを指定していないときは 180dpi が採用されます。解像度 180dpi 以外のプリンタをお使いの方は、print.cfg にこのオプションを加えておくとよいでしょう。本書添付のインストーラを使用した場合は、すでに正しい値が入っています。

LP-3000 用 print.cfg の設定例を挙げます。

List 2-4● LP–3000 用 print.cfg

```
 1: -remark= ＬＰ－３０００用 print.cfg by Ritchy:
 2: -remark=        modified by T.Hilano for Raster_Output
 3: -width=2360
 4: -height=3376
 5: -xOffset=200
 6: -yOffset=200
 7: -dpi=300
 8: -pinBytes=1
 9: -init=\x1bz%0%0\x1d0;0.24muE\x1d8;0lpP\x1d0;300;300drE\x1d
    0boP
10: -RasterOutPutOrder=xygXY
11: -Raster
12: -Raster_yPos=\x1d%4DY
13: -Raster_xPos=\x1d%4DX
14: -Raster_ySize=%4D;0bi{I
15: -Raster_xSize=%4D;
16: -CRLF=\r\n
17: -FF=\f
18: -graphic=\x1d%5D;
19: -start=
20: -repeat=
```

-mag　拡大率指定

書式 1：　-mag=[num]

書式 2：　-mag=half

機能：　拡大率を指定します。

解説：　この指定に従って、印字拡大率が変化します。このオプションの指定がない場合、dvi ファイルに書き込まれている値が採用されます。以下に、各指定値に対応する実際の拡大率とフォントの dpi 値を挙げます。「フォント」の項目は、-dpi=180 と指定した場合、拡大後、どのフォントが主に使われるかを示したものです。

指定	拡大率	フォント
0	1.000 倍	180dpi
half	1.096 倍	197dpi
1	1.200 倍	216dpi
2	1.440 倍	259dpi
3	1.728 倍	311dpi
4	2.074 倍	373dpi
5	2.488 倍	448dpi

-timer　キースキャン・タイマ・インターバル指定

書式：　-timer=[num]

機能：　キースキャンのためのタイマ・インターバルを指定します。

解説：　デフォルト値は 60 です。[num] の値を小さくすると、キースキャンは速くなります。

以下に、キースキャンを少し速くしてみる場合の指定例を挙げます。

```
A> print -timer=50 sample
```

-TRAM　テキスト RAM 使用宣言

書式：　-TRAM

機能：　テキスト RAM も cell 領域に使用します。

解説：　テキスト RAM も cell 領域に使用したい場合に指定します。
プリンタドライバのコンフィギュレーションファイルに追加しておくとよいでしょう。

List 2-5● -TRAM の設定例

```
 1: -remark= ＨＧ８００用 print.cfg
 2: -dpi=180
 3: -MSBisUpper
 4: -TRAM
 5: -pinBytes=3
 6: -init=\e@\x18\eA\x08
 7: -CRLF=\r
 8: -FF=\f
 9: -xOffset=135
10: -graphic=\e*\x27%2i
11: -start=
12: -repeat=
13: -height=1944
```

-GRAM　グラフィック RAM 使用宣言

書式：　-GRAM

機能：　グラフィック RAM も cell 領域に使用します。

解説：　グラフィック RAM も cell 領域に使用したい場合に指定します。グラフィック RAM が他の目的に使用されていた場合はエラーを表示して終了します。

```
A> print -GRAM sample
X680x0 TeX Printer Driver Ver 2p09a Copyright 1989,90,
91,92,93 by Ｅｘｔ (T.Kawamoto)
       Tpic Specials support Ver 1.00a Copyright 1993
 by TSG+SHIMA
       RasterOutput Devices support Ver 1.00 Copyright
 1992, 1993 by T. Hilano

Program Stop : graphic ＲＡＭは使用できません。
A>
```

この表示が出た場合には、グラフィック RAM を専有しているアプリケーション[12]を外し、使用できるようにします。もしそのアプリケーションを外したくなければ、残念ですが、プリンタドライバでのグラフィック RAM の使用 (-GRAM) をあきらめるしかありません。

12)最も可能性の高いものは、グラフィック RAM ディスク でしょう。

-landscape, -vertical, -turnRight
横置き (縦方向) 印字指定 (右回転)

書式 1： -landscape

書式 2： -vertical

書式 3： -turnRight

機能： 横置き (縦方向) 印字を指定します。

解説： プリンタに出力する際、ビットマップデータを時計回りに 90 度回転し、縦方向に印字します。

A4 用紙を横位置にして印刷する場合に使います。landscape.sty と組み合わせて使用してください。

List 2-6 ● -landscape の使用例 — sample.tex

```
1:  \documentstyle[landscape]{jreport}
2:                    :
3:                    :
```

List 2-6 を LaTeX にかけます。

```
A>latex sample
```

これを出力する際に “-landscape” オプションをつけます。

```
A>print -landscape sample
```

なお、-vertical は以前のバージョンのプリンタドライバとの互換性を考えて残してありますが、-landscape と同じ意味です。また、-turnRight については、次のページで説明する-turnLeft オプションとセットにする意味で設けています。-turnRight も -landscape と同じ意味です。

-turnLeft　横置き (縦方向) 印字指定 (左回転)

書式：　-turnLeft

機能：　横置き (縦方向) 印字を指定します。

解説：　プリンタに出力する際、ビットマップデータを時計回りとは逆に 90 度回転し、縦方向に印字します。

A4 用紙を横位置にして印刷する場合に使います。landscape.sty と組み合わせて使用してください。

List 2-7 ● -turnLeft の使用例 — sample.tex

```
1:  \documentstyle[landscape]{jreport}
2:                          :
3:                          :
```

List 2-7 を LaTeX にかけます。

```
A>latex sample
```

これを出力する際に “-turnLeft” オプションをつけます。

```
A>print -turnLeft sample
```

-width　バッファ横ドット数指定

書式： -width=[num]

機能： バッファの横幅をドット単位で指定します。

解説： デフォルトは 1440 です。普通は、プリント幅をドット単位で指定します。縦書き指定がある場合は、プリント用紙 1 ページの長さをドット単位で指定します。8 の倍数で指定してください。

List 2-8● -width の設定例

```
 1: -remark=ＰＣ－ＰＲ１０１Ｅ２用 print.cfg　ｂｙ　どおお
 2: -width=1320
 3: -height=1800
 4: -xOffset=-130                 -remark=適当に変えてください
 5: -yOffset=-60
 6: -init=\ec\x31\eT\x31\x37
 7: -CRLF=\r\n
 8: -FF=
 9: -graphic=\eJ%4d
10: -repeat=\eU%4d           -remark=；これ、大丈夫かな？
11: -start=
12: -prBufSize=1024
```

-height　バッファ縦ドット数指定

書式： -height=[num]

機能： バッファの縦幅をドット単位で指定します。

解説： デフォルトは 1980 です。普通は、プリント用紙 1 ページの長さをドット単位で指定します。縦書き指定がある場合は、プリント幅をドット単位で指定します。8 の倍数で指定してください。

List 2-9● -height の設定例

```
 1: -remark=EPSON_AP-900　by　諸は郎
 2: -prBufSize=512
 3: -GRAM
 4: -TRAM
 5: -width=2880
 6: -height=3920
 7: -dpi=360
 8: -MSBisUpper
 9: -pinBytes=6
10: -init=\eA\x08
11: -CRLF=\r
12: -FF=\f
13: -graphic=\e*\x48%2i
14: -start=
15: -repeat=
```

-xOffset　左上座標指定 (X 座標)

書式： -xOffset=[num]

機能： プリンタ用紙に対して書き込む位置を指定します。

解説： 次ページで説明する “-yOffset” とセットで使用します。座標系は、画面左上を原点とし、右が X 軸の正の方向、下が Y 軸の正の方向です。これらのオプションで、どの位置から書き込みを始めるかを指定します。たとえば、“-xOffset=0” と指定します。

> 次ページで説明する -yOffset とセットで使用します。
> 座標系は、画面左上を原点とし、右が X 軸の正の方向、下が Y 軸の正の方向です。これらのオプションで、どの位置から書き込みを始めるかを指定します。

これを “-xOffset=100” と指定すると、100 ドット分、右方向にずれます。

> 次ページで説明する -yOffset とセットで使用します。
> 座標系は、画面左上を原点とし、右が X 軸の正の方向、下が Y 軸の正の方向です。これらのオプションで、どの位置から書き込みかを指定します。

逆に負の値を指定すると、左方向にずれます。

デフォルトは、紙の中央が画面の中央にくるような値が採用されます。

-yOffset　左上座標指定（Y 座標）

書式：　-yOffset=[num]

機能：　プリンタ用紙に対して書き込む位置を指定します。

解説：　前ページで説明した“-xOffset”とセットで使用します。座標系は、画面左上を原点とし、右が X 軸の正の方向、下が Y 軸の正の方向です。これらのオプションで、どの位置から書き込みを始めるかを指定します。たとえば、“-yOffset=0”と指定します。

前ページで説明した -xOffset とセットで使用します。
座標系は、画面左上を原点とし、右が X 軸の正の方向、下が Y 軸
の正の方向です。これらのオプションで、どの位置から書き込みを始める
かを指定します。

これを“-yOffset=100”と指定すると、100 ドット分、下方向にずれます。

前ページで説明した -xOffset とセットで使用します。
座標系は、画面左上を原点とし、右が X 軸の正の方向、下が Y 軸

逆に負の値を指定すると、上方向にずれます。
デフォルトは、紙の中央が画面の中央にくるような値が採用されます。

-pk2fontman　FONTMAN Ver.2 の変換ファイル指定

書式： -pk2fontman=[string]

機能： FONTMAN Ver.2 の変換ファイル指定。

解説： FONTMAN Ver.2 を利用する場合、print.p2m ファイルを必要とします。FONTMAN Ver.2 は、デフォルトでは、カレントディレクトリの print.p2m、 あるいは %TEXHOME% の print.p2m を参照しますが、このオプションを使えば変更できます。この変更は、環境変数 PRINT.P2M または PRINT_P2M でも設定可能です。オプションと環境変数の両方を指定した場合は、オプションの指定のほうが優先されます。ファイル名をフルパスで指定してください。詳しくは第 1.4 節「PRINT 環境変数」(p.12) を参照してください。

なお、FONTMAN Ver.2 は、本書には添付されていません。

-pk3fontman　FONTMAN Ver.3 の変換ファイル指定

書式： -pk3fontman=[string]

機能： FONTMAN Ver.3 の変換ファイル指定です。

解説： FONTMAN Ver.3 を利用する場合、print.p3m ファイルを必要とします。フォントマネージャの Ver. 3 はデフォルトでは、カレントディレクトリの print.p3m、あるいは %TEXHOME% の print.p3m を参照しますが、このオプションを使えば変更することができます。この変更は、環境変数 PRINT.P3M または PRINT_P3M でも設定可能です。オプションと環境変数の両方を指定した場合は、オプションの指定のほうが優先されます。

ファイル名をフルパスで指定してください。詳しくは第 1.4 節「PRINT 環境変数」(p.12) を参照してください。

-fontmanMaxSize　フォントマネージャの最大フォントサイズ

書式： `-fontmanMaxSize=[num]`

機能： フォントマネージャの最大フォントサイズを指定します。

解説： FONTMAN の Ver.2 あるいは Ver.3 を利用すると、かなり大きなサイズの漢字フォントまで利用することができます。フォントマネージャとのインターフェースでは、フォントのバッファを必要なサイズだけ確保しなければならないので、このオプションで指定します。デフォルトは 512 ドットですが、普通はこれで十分です。

-layout　レイアウト印字指定

書式： `-layout`

機能： レイアウト印字を指定します。

解説： すべてのフォントのすべての文字がボックスで印字されます。印字速度が若干速くなりますが、たいした効果は期待できません。

-subst　代用印字方法指定

書式：　-subst=[string]

機能：　代用印字方法を指定します。

解説：　フォントファイルがない場合の動作を指定します。[string] は、box, cmrmin, nofont のうちから選んでください。それぞれの動作は、以下のとおりです。

(1)　-subst=box

フォントファイルがない場合、ボックス印字します。ただし、tfm ファイルがないと、ボックス印字をすることはできません。その場合、ダミーフォントとなります。

(2)　-subst=cmrmin

フォントファイルがない場合、cmr[13]か min[14]で代用します。この場合、フォントの領域を示すボックスは正確ではなくなります。なお、これらのフォントがない場合は、代用印字することはできません。その場合、ダミーフォントとなります。

この指定で代用印字をした場合、“-info” で見ると、“check sum error” がよく表示されます。しかし、無視してかまいません。

(3)　-subst=nofont

フォントファイルが見つからなかった場合、ダミーフォントとなります。

なお、上述のいかなる場合でもダミーフォントが見つかった場合は、途中で動作を打ち切ります。該当フォントを作成したうえでリトライしてください。

13)英語フォントの場合。

14)日本語フォントの場合。

-dump　ダンプ出力

書式：　-dump=[string]

機能：　ダンプ出力を指定します。

解説：　プリンタに出力するかわりに、バイナリデータをファイルに出力します。

```
A>copy ファイル名 LPT
```

以上の操作でプリンタに出力できる形式でファイルに出力しますから、いったんファイルに保存しておいて、後日プリンタに出力することができます。

-MSBisUpper　印字ヘッド方向指定

書式：　-MSBisUpper

機能：　印字ヘッドの方向を指定します。

解説：　多くのプリンタ[15]は、プリンタの印字ヘッドピンの上方向に LSB がきます。しかし、なかにはシャープの CZ シリーズのようにビット並びが反対[16]のものがあります。そういう場合、本オプションを指定します。

ここでちょっと寄り道をして、プリンタの印字ヘッドピンの並びについて説明しておきます。プリンタをコンピュータで制御するために、普通、プリンタの印字ヘッドピンは 8 つごとに区切ってバイト単位にまとめられています。1 バイトに含まれる 8 つのビット並びを正確に把握していないと、数ミリ幅の帯状にスライスされたイメージがプリンタに出力されてしまいます。並び方は、上下が入れ替わっているかどうかという違いしかありません。それは“プリンタの上方に MSB がくるか LSB がくるか”と表現されます。あなたのプリンタがどちらであるかはわかりにくいと思いますが、見分け方のひとつとして，プリンタがシャープ製かどうかという方法があります。シャープ製なら、確実に -MSBisUpper を指定すべきプリンタだからです。その他のメーカについては、それぞれのプリンタのマニュアルを参照してください。

そして万が一、“数ミリ幅の帯状にスライスされたイメージ”が出力されたら、-MSBisUpper を再調整してください。

15)たとえば、NEC の PR シリーズ等。

16)バイトの MSB がプリンタの上方にくる。

List 2-10 ● -MSBisUpper の設定例

```
 1:  -remark= ＣＺ－８ＰＣ３用 print.cfg by ＪｕｎＫ
 2:  -dpi=180
 3:  -MSBisUpper
 4:  -pinBytes=3
 5:  -init=\ec1\e%%9\x10\x20\x08
 6:          -remark= 美しくないけどよしとしよう by じぱんぐ
 7:  -CRLF=\r\n
 8:  -FF=\f          -remark= corrected by じぱんぐ 89/12/11
 9:  -graphic=\eJ%2m
10:  -start=\e\x10%4d
11:  -repeat=
```

-MSBisLeft 印字ヘッド方向指定

書式： -MSBisLeft

機能： 印字ヘッドの方向を指定します。

解説： PR2000/4 のように、プリンタの印字ヘッドのビットの並びが横に並んでいるときに、本オプションを使います。

List 2-11 ● -MSBisLeft の設定例

```
 1: -remark= ＰＲ２０００／４用 print.cfg by ＳＡＳＡ
 2: -remark= Ａ４縦書き用
 3: -dpi=400
 4: -TRAM
 5: -GRAM
 6: -width=3040
 7: -height=4512
 8: -MSBisLeft
 9: -pinHeights=1
10: -init=\x1bc1\x1cd240.\x1cfPA4.\x1c<1/400,i.\x1ce0,0.
11: -CRLF=\x1ce0,1,,B,B.
12: -FF=\f
13: -graphic=\x1ci%4d,1,0,1/1,1/1,%3D,400.
14: -start=
15: -repeat=
```

残念ながら、本オプションを指定した場合、縦方向への印字ができなくなります。しかし、PR2000/4 にかぎっていえば、-MSBisLeft を外してラスタプリンタ制御系オプションを駆使すると、縦方向への印字が可能になります。

List 2-12 ● PR2000/4 で縦方向への印字を行う例

```
 1: -remark= ＰＲ２０００／４用 print.cfg by ＳＡＳＡ
    :Modified_by_T.Hilano
 2: -remark= Ａ４縦書き用
 3: -dpi=400
 4: -TRAM
 5: -GRAM
 6: -width=3040
 7: -height=4512
 8: -Raster
 9: -RasterOutPutOrder=xyXYg
10: -Raster_xPos=\x1ce%4D
11: -Raster_yPos=,%4D.
12: -Raster_xSize=\x1ci%4D
13: -Raster_ySize=,%4D
14: -pinHeights=1
15: -init=\x1bc1\x1cd240.\x1cfPA4.\x1c<1/400,i.\x1ce0,0.
16: -CRLF=\x1ce0,1,,B,B.
17: -FF=\f
18: -graphic=,0,1/1,1/1,%4D,400.
19: -start=
20: -repeat=
```

-pinBytes　印字ヘッドのピン数指定

書式：　-pinBytes=[num]

機能：　印字ヘッドのピン数を指定します。

解説：　[num] には、印字ヘッドのピン数/8 を指定してください。

-pinHeights　印字ヘッドのピン数指定

書式：　-pinHeights=[num]

機能：　印字ヘッドのピン数を指定します。

解説：　[num] で印字ヘッドのピン数を指定します。-MSBisLeft を指定した場合は、-pinBytes ではなく、-pinHeights を指定してください。

-prBufSize　プリンタバッファの大きさ指定

書式： -prBufSize=[num]

機能： プリンタバッファの大きさを指定します。

解説： プリンタスプールに使用するバッファの大きさを K バイト単位で指定します。1 以上を指定してください。デフォルトは 256 です。プリンタの速度が X680x0 の描画速度に比べて遅い場合、このバッファの値を大きくするとよいでしょう。反対に、プリンタの速度が X680x0 よりも速い場合には、最小値の 1 を指定してください。

-init　プリンタ初期化のコード列指定

書式：　-init=[string]

機能：　プリンタ初期化のコード列を指定します。

解説：　1ページの始まりごとに [string] に指定したコード列をプリンタに送ります。プリンタに出力するエスケープ・シーケンス等のコード列を指定します。ここで指定できるコードの詳しい説明は、第2.4.3項「-switch=[string] で使用できるコード」(p.91) を参照してください。デフォルトは、-init=\ec1\eM\sG\eT24 です。

List 2-13 ● -init の設定例

```
 1: -remark= ＨＰ－ＬａｓｅｒＪｅｔ４用 print.cfg by HIDE
 2: -dpi=600
 3: -width=4672
 4: -height=6400
 5: -pinHeights=1
 6: -MSBisLeft
 7: -CRLF=\n
 8: -FF=\f
 9: -xOffset=400
10: -yOffset=428
11: -init=\eE\e&u600D\e*t600R\e&l.08C\e&k2G\e&f0y0X\e*p-1Y\e*r
    b1A\e&f1X
12: -graphic=\e&f2X\e*b%3DW
13: -start=\e*p%4dX
14: -repeat=
```

List 2-13 の 11 行目を見てください。-init の行の設定により、1ページの始まりごとに「エスケープコード＋Ｅ＋エスケープコード＋&＋ｕ＋６＋…」という長いコード列をプリンタに出力します。

-CRLF　プリンタ改行のコード列指定

書式：　-CRLF=[string]

機能：　プリンタ改行のコード列を指定します。

解説：　改行ごとに [string] で指定したコード列を送ります。プリンタに出力するエスケープ・シーケンス等のコード列を指定します。ここで指定できるコードの詳しい説明は、第2.4.3項「-switch=[string] で使用できるコード」(p.91) を参照してください。

デフォルトは -CRLF=\r\n (改行復帰) です。

List 2-14● -CRLF の設定例

```
 1: -remark=  ＰＲ２０００／４用  print.cfg  by  ＳＡＳＡ
 2: -remark=  Ｂ５縦書き用
 3: -dpi=400
 4: -TRAM
 5: -GRAM
 6: -width=2608
 7: -height=3888
 8: -MSBisLeft
 9: -pinHeights=1
10: -init=\x1bc1\x1cd240.\x1cfPB5.\x1c<1/400,i.\x1ce0,0.
11: -CRLF=\x1ce0,1,,B,B.
12: -FF=\f
13: -graphic=\x1ci%4d,1,0,1/1,1/1,%3D,400.
14: -start=
15: -repeat=
```

List 2-14 の 11 行目を見てください。-CRLF の行の設定により、改行には、「0x1c ＋ e ＋ 0 ＋ , ＋ 1 ＋ , ＋ , ＋ B ＋ , ＋ B ＋ .」を使います。

-extraCRLF　空行改行時の追加コード列指定

書式：　-extraCRLF=[string]

機能：　空行改行時の追加コード列を指定します。

解説：　空行改行時に、改行コード列の直前に [string] で指定したコード列を追加します。空行の出力時に通常の改行だけでは不十分な場合、本オプションで不足分を追加します。プリンタに出力するエスケープ・シーケンス等のコード列を指定します。ここで指定できるコードの詳しい説明は、第 2.4.3 項「-switch=[string] で使用できるコード」(p.91) を参照してください。デフォルトは -extraCRLF= (何もしない) です。

HP-DJ505J のコンフィギュレーションファイルは、本オプションを使って空行の調整を行います。

List 2-15 ● -extraCRLF の設定例

```
 1:  -remark=HP_DJ505J_Config_by_AIR
 2:  -width=2400
 3:  -height=3300
 4:  -dpi=300
 5:  -MSBisLeft
 6:  -pinHeights=1
 7:  -init=\e@\e\x5F\x41%0
 8:  -CRLF=\r
 9:  -extraCRLF=\e\x5F\x4A\x01%0
10:  -FF=\f
11:  -graphic=\e\x5F\x57%2I
12:  -start=
13:  -repeat=
14:  -relative=
```

List 2-15 の 8、9 行目を見てください。-CRLF の行の設定により、通常行は復帰 0x0d を出力するのみですが、-extraCRLF の行の設定により、空行を出力する場合はかわりに「エスケープコード＋_＋ J ＋ 0x01 ＋ 0x00 ＋ 0x0d」の順に出力します。

-FF　プリンタ改ページのコード列指定

書式：　-FF=[string]

機能：　プリンタ改ページのコード列を指定します。

解説：　改ページ時に [string] で指定したコード列を追加します。プリンタに出力するエスケープ・シーケンス等のコード列を指定します。ここで指定できるコードの詳しい説明は、第2.4.3項「-switch=[string] で使用できるコード」(p.91) を参照してください。デフォルトは -FF=\f\d です。

List 2-16 ● -FF の設定例

```
 1:  -remark= MJ-500 print.cfg by 山川直己
 2:  -dpi=360
 3:  -MSBisUpper
 4:  -TRAM
 5:  -pinBytes=6
 6:  -init=\e@\x18\eA\x08
 7:  -CRLF=\r\n
 8:  -FF=\f
 9:  -xOffset=135
10:  -graphic=\e*\x48%2i
11:  -start=
12:  -repeat=
13:  -relative=\e\\%2/2i
14:  -xOffset=300
15:  -yOffset=0
16:  -PrBufSize=32
```

List 2-16 の 8 行目を見てください。-FF の行の設定により、改ページする際には 0x0c を出力します。

-graphic　ドットグラフィック指定のコード列指定

書式：　-graphic=[string]

機能：　ドットグラフィック指定のコード列を指定します。

解説：　プリンタドライバが作成したビットマップデータ1行分をプリンタに送る際に使用するコード列です。[string] で指定されるコード列をすべて出力後、ビットイメージデータが出力されます。デフォルトは -graphic=\eJ%4d です。

ビットマップ出力のために、-graphic, -start, -relative, -repeat の 4 つのオプションが用意されています。この 4 つのなかでは、-graphic が最も基本的で、かつ、最も効率が悪い方法です。詳しくはプリンタのマニュアルを参照してください。プリンタのマニュアルには、「ドットグラフィック指定」といった用語で記述されています。プリンタのマニュアルをよく読んで「ドットグラフィック指定」という用語があれば、本オプションで指定してください。

[string] に使用可能なコードについての詳細は、第 2.5.3 項「-switch=[string] で使用できるコード」(p.91) を参照してください。

List 2-17● -graphic の設定例

```
 1:  -remark=EPSON_AP-900   by   諸は郎
 2:  -prBufSize=512
 3:  -GRAM
 4:  -TRAM
 5:  -width=2880
 6:  -height=3920
 7:  -dpi=360
 8:  -MSBisUpper
 9:  -pinBytes=6
10:  -init=\eA\x08
11:  -CRLF=\r
12:  -FF=\f
13:  -graphic=\e*\x48%2i
14:  -start=
15:  -repeat=
```

List 2-17 の 13 行目を見てください。-graphic の行の設定により、「エスケープコード＋*＋H」に続き、リトルエンディアン[17]で出力し、最後にビットマップデータがその長さだけ続くことを意味します。

17)インテルの CPU が採用している順番で、2 バイトのうち、下位バイトをまず先に、上位バイトをそのあとに出力する方法。

-start　ドット単位印字開始位置指定のコード列指定

書式：　-start=[string]

機能：　ドット単位印字開始位置指定のコード列を指定します。

解説：　プリンタドライバが作成したビットマップデータ 1 行分をプリンタに送る際に、空白をスキップするためのコード列です。デフォルトは -start=\eF%4d です。

ビットマップをベタ出力する際に空白部分を省略すると、プリンタへの転送容量が減るため、高速化できます。本オプションは、そういった用途に使われるコード列を指定するものです。

プリンタの 1 行の印字開始位置を指定する目的で、プリンタにビットマップ列を出力する先頭絶対位置を指定できる機能が備わっていることがあります。プリンタのマニュアルには、「ドット単位印字開始位置指定」といった用語で記述されています。詳しくはプリンタのマニュアルを参照してください。

プリンタのマニュアルをよく読んで、このような指定のしかたが可能なことを確かめてから、本オプションを指定してください。もし不幸にも本機能がない場合には、[string] を省略して -start= と書きます。

使用可能なコードについての詳細は、第 2.4.3 項「-switch=[string] で使用できるコード」(p.91) を参照してください。

List 2-18 ● -start の設定例

```
 1: -remark= ＦＭ－ＰＲ４５Ｉ用 print.cfg
 2: -remark=by_fatman
 3: -dpi=180
 4: -MSBisUpper
 5: -pinBytes=3
 6: -init=\ec\eQ\x32\x3b\x31\x35\x20G
 7: -remark=                        reset&LFpicth
 8: -CRLF=\n
 9: -FF=\f
10: -graphic=\eQ%4d\x20W
11: -start=\e[%4d\x60                         -remark=absolute
12: -relative=\e[%4da                         -remark=relative
13: -width=2448
14: -height=1980
```

List 2-18 の 11 行目を見てください。-start の行の設定により、「エスケープコード＋［」に続き、位置を最大 4 桁までの ASCII コードで表した十進数で出力し、最後に ‘ を出力することで、先頭絶対位置を指定しています。実際のビットマップの出力は -graphic で指示されたとおりに行われます。

-relative　ドット単位印字位置相対指定のコード列指定

書式： -relative=[string]

機能： ドット単位印字位置相対指定のコード列を指定します。

解説： プリンタドライバが作成したビットマップデータ 1 行分をプリンタに送る際に、空白をスキップするためのコード列です。デフォルトは -relative= です。ビットマップをベタ出力する際に空白部分を省略すると、プリンタへの転送容量が減るため、高速化できます。本オプションは、そういった用途に使われるコード列を指定するものです。

プリンタの 1 行目の印字開始位置を指定するために、プリンタにはビットマップ列を出力する先頭相対位置を指定できる機能が備わっていることがあります。プリンタのマニュアルには、「ドット単位印字位置相対指定」といった用語で記述されています。詳しくはプリンタのマニュアルを参照してください。

-start と同じ目的で使用されます。ただし、-relative は、ビットマップ列を出力する先頭位置を、以前出力した先頭位置からの相対位置で指定するコードが備わっているときに使います。

プリンタのマニュアルをよく読んで、このような指定のしかたが可能なことを確かめてから、本オプションを指定してください。もし不幸にも本機能がない場合には、[string] を省略して -relative= と書きます。

使用可能なコードについての詳細は、第 2.4.3 項「-switch=[string] で使用できるコード」(p.91) を参照してください。

List 2-19 ● -relative の設定例

```
 1: -remark= ＢＪ｜３０Ｊ用 print.cfg writen by あるたさろす・Ｒ・
    ミオラルオン
 2: -dpi=360
 3: -pinBytes=6
 4: -init=\eM\sG\sC\eT24
 5: -CRLF=\r\n
 6: -FF=\f
 7: -MSBisUpper
 8: -graphic=\x1cCB\x04\x04%2i
 9: -relative=\eU%4/2d%0%0%0 -remark=added_by_Ｓａｓａｆｕ
10: -start=
11: -repeat=
12: -width=2880
13: -height=3960
```

List 2-19 の 9 行目を見てください。-relative の行の設定により、エスケープコード＋U に続き、相対位置を 2 で割ったものを、最大 4 桁までの ASCII コードで表した十進数で出力し、続けて 3 つの 0x00 を出力することで相対位置を指定しています。実際のビットマップの出力は -graphic で指示されたとおりに行われます。

-repeat　グラフィックリピート指定のコード列指定

書式： -repeat=[string]

機能： グラフィックリピート指定のコード列を指定します。

解説： プリンタドライバが作成したビットマップデータ 1 行分をプリンタに送る際に、空白をスキップするためのコード列です。デフォルトは -repeat=\eU%4d です。ビットマップをベタ出力する際に空白部分を省略すると、プリンタへの転送容量が減るため、高速化できます。本オプションは、そういった用途に使われるコード列を指定するものです。プリンタに、同じビットマップデータを繰り返し転送する機能が備わっていることがあります。プリンタのマニュアルには、「グラフィックリピート指定」といった用語で記述されています。詳しくはプリンタのマニュアルを参照してください。

-start, -relative とは違って出力位置が指定できなくても、リピート指定ができれば、これらと同じようにプリンタへの転送容量を削減できます。[string] で指定されるコード列をすべて出力後、リピートすべきビットイメージデータが出力されます。プリンタのマニュアルをよく読んで、このような指定のしかたが可能なことを確かめてから、本オプションを指定してください。もし不幸にも本機能がない場合には、[string] を省略して -repeat= と書きます。使用可能なコードについての詳細は、第 2.4.3 項「-switch=[string] で使用できるコード」(p.91) を参照してください。

List 2-20 ● -repeat の設定例

```
 1: -remark= Laser Shot emulation (2/3 mode)用 (LBP180L.cfg)
    by ＭＡＳＡ
 2: -remark= １ ５ ９ ... と ３ ７ 11 ... の page で使います。
 3: -dpi=180
 4: -pinBytes=3
 5: -height=1960
 6: -width=2560
 7: -turnLeft
 8: -init=\x1bc1\x1bM\x1bT18\x1bd2
 9: -CRLF=\r\n
10: -FF=\x0d
11: -xOffset=1340
12: -yOffset=110
13: -graphic=\eJ%4d
14: -start=\eF%4d
15: -repeat=\eU%4d
```

List 2-20 の 15 行目を見てください。“-repeat” の行の設定により、「エスケープコード＋U」に続き、リピート回数を最大 4 桁までの ASCII コードで表した十進数で出力し、続いてリピートすべきコードを 1 バイト出力するように指定しています。実際のビットマップの出力は、-graphic で指示されたとおりに行います。

-Raster　ラスタスキャン出力指定

書式：　-Raster

機能：　ラスタスキャン出力することを指定します。

解説：　-Raster オプションを指定すると、次のオプションを指定しても無視されます。

- ○ -vertical
- ○ -MSBisUpper
- ○ -pinBytes
- ○ -CRLF
- ○ -start
- ○ -relative
- ○ -repeat

List 2-21 ● -Raster の設定例

```
 1:  -remark= ＬＰ－３０００用 print.cfg by Ritchy:
 2:  -remark=           modified by T.Hilano for Raster_Output
 3:  -width=2360
 4:  -height=3376
 5:  -xOffset=200
 6:  -yOffset=200
 7:  -dpi=300
 8:  -pinBytes=1
 9:  -init=\x1bz%0%0\x1d0;0.24muE\x1d8;01pP\x1d0;300;300drE\x1
     d0boP
10:  -OrderPos=x
11:  -OrderSize=gxy
12:  -Raster
13:  -Raster_yPos=\x1d%4DY
14:  -Raster_xPos=\x1d%4DX
15:  -Raster_ySize=%1D;0bi{I
16:  -Raster_xSize=%3D;
17:  -CRLF=\r\n
18:  -FF=\f
19:  -graphic=\x1d%4D;
20:  -start=
21:  -repeat=
```

-MH　MH 符号圧縮指定

書式：　-MH

機能：　ラスタスキャンのデータを MH 符号で圧縮して転送する指定をします。

解説：　MH 符号圧縮指定は、-Raster を指定していないと無効です。MH 符号は 1 本の主走査線の画素数が 2560 以下の白黒 2 値の画像信号を対象としています。-width で指定する値は、この値より小さくしてください[18]。

18) これ以上の値を指定した場合には、安定した動作は保証されません。

-EveryRaster　出力ラスタ数指定

書式：　-EveryRaster=[num]

機能：　[num] 個のラスタごとに出力するように指定します。

解説：　-EveryRaster オプションは、-MH 指定時には無視されます。MH 符号圧縮をしない場合には、空白のラスタ群にはさまれた部分をひとまとめにしてプリンタに出力します。その際、送るビットイメージの前後にある空白は除きます。したがって、空白のラスタがあまりなく、その間でラスタの長さが大きく変化する場合等には有効です[19]。

19) たとえば、スタイルオプション (j)twocolumn を指定しているときなど。

-RasterOutPutOrder　ラスタ情報出力順序指定

書式： -RasterOutPutOrder=[string]

機能： 出力するラスタイメージデータの位置や大きさを指示する順序を定義します。

解説： [string] には、次に挙げる文字を組み合わせて指定します。

文字	意味	関連オプション
x	x 方向（横）の位置	-Raster_xPos
y	y 方向（縦）の位置	-Raster_yPos
X	ラスタスキャンの x 方向の大きさ	-Raster_xSize
Y	ラスタスキャンの y 方向の大きさ	-Raster_ySize
g	送るべきバイト数	-graphic

[string] に記述した「文字」の順序に従って、「関連オプション」で指定したコード列が出力されます。なお、すべての文字を書く必要はありません。

List 2-22 ● -RasterOutPutOrder の設定例

```
 1:  -remark= ＬＰ－３０００用 print.cfg by Ritchy:
 2:  -remark=           modified by T.Hilano for Raster_Output
 3:  -width=2360     -height=3376
 4:  -xOffset=200    -yOffset=200
 5:  -dpi=300        -pinBytes=1
 6:  -init=\x1bz%0%0\x1d0;0.24muE\x1d8;01pP\x1d0;300;300drE\x1
     d0boP
 7:  -RasterOutPutOrder=xygXY
 8:  -Raster
 9:  -Raster_yPos=\x1d%4DY
10:  -Raster_xPos=\x1d%4DX
11:  -Raster_ySize=%4D;0bi{I
12:  -Raster_xSize=%4D;
13:  -CRLF=\r\n
14:  -FF=\f
15:  -graphic=\x1d%5D;
16:  -start=
17:  -repeat=
```

List 2-22 の例の場合、7 行目は、ビットマップをラスタ出力する際、「x 方向の位置」、「y 方向の位置」、「送るべきバイト数」、「ラスタイメージの x 方向の大きさ」、そして、「ラスタイメージの y 方向の大きさ」の順に送ることを表しています。

「x 方向の位置」の出力コードは、10 行目の -Raster_xPos の記述に従います。「y 方向の位置」の出力コードは、9 行目の -Raster_yPos の記述に従います。「送るべきバイト数」の出力コードは、15 行目の -graphic の記述に従います。「ラスタイメージの x 方向の大きさ」の出力コードは、12 行目の -Raster_xSize の記述に従います。「ラスタイメージの y 方向の大きさ」の出力コードは、11 行目の -Raster_ySize の記述に従います。

-Raster_xPos　ラスタスキャン x 方向位置のコード列指定

書式： -Raster_xPos=[string]

機能： ラスタスキャン x 方向位置のコード列を指定します。

解説： -Raster_xPos では、[string] にラスタスキャンデータの x 方向出力位置をドット単位で指定するコード列を書きます。

次に説明する -Raster_yPos と組み合わせて使います。

p.88 の List 2-22 の例の 10 行目を見てください。-Raster_xPos の行の設定により、x 方向の位置をバイト数で数えたものを、最大 4 桁までの ASCII コードで表した十進数で出力し、続けて X を出力します。

-Raster_yPos　ラスタスキャン y 方向位置のコード列指定

書式： -Raster_yPos=[string]

機能： ラスタスキャンの y 方向位置のコード列を指定します。

解説： -Raster_yPos では、[string] にラスタスキャンデータの y 方向出力位置をドット単位で指定するコード列を書きます。

上の -Raster_xPos と組み合わせて使います。

p.88 の List 2-22 の例の 9 行目を見てください。-Raster_yPos の行の設定により、y 方向の位置をバイト数で数えたものを、最大 4 桁までの ASCII コードで表した十進数で出力し、続けて Y を出力します。

-Raster_xSize　ラスタスキャン x 方向サイズのコード列指定

書式： -Raster_xSize=[string]

機能： ラスタスキャン x 方向サイズのコード列を指定します。

解説： -Raster_xSize では、[string] に、ラスタスキャンデータの x 方向サイズをドット単位で指定するコード列を書きます。

次の -Raster_ySize と組み合わせて使います。

p.88 の List 2-22 の例の 12 行目を見てください。-Raster_xSize の行の設定により、x 方向サイズをバイト数で数えたものを、最大 4 桁までの ASCII コードで表した十進数で出力します。

-Raster_ySize　ラスタスキャン y 方向サイズのコード列指定

書式： -Raster_ySize=[string]

機能： ラスタスキャン y 方向サイズのコード列を指定します。

解説： -Raster_ySize では、[string] に、ラスタスキャンデータの y 方向サイズをドット単位で指定するコード列を書きます。

上の -Raster_xSize と組み合わせて使います。

p.88 の List 2-22 の例の 11 行目を見てください。-Raster_ySize の行の設定により、y 方向サイズをバイト数で数えたものを、最大 4 桁までの ASCII コードで表した十進数で出力します。その後、「; + 0 + b + i +{+ I」を続けて出力します。

2.4.3 -switch=[string] で使用できるコード

プリンタドライバで使用できるオプションのうち、次のものは引数として文字列を指定します。

○ **汎用プリンタ制御系**

-init, -CRLF, -extraCRLF, -FF, -graphic, -start, -relative, -repeat

○ **ラスタプリンタ制御系**

-Raster_xPos, -Raster_yPos, -Raster_xSize, -Raster_ySize

そこに指定できるコードを以下に説明します。なかでも次に示す 8 つのオプションは、ビットマップデータの位置・サイズ等の情報をともなうので、次に説明するような特別な記法が用意されています。

○ **汎用プリンタ制御系**

-graphic, -start, -relative, -repeat

○ **ラスタプリンタ制御系**

-Raster_xPos, -Raster_yPos, -Raster_xSize, -Raster_ySize

◆ %?d, %?D, %?m, %?M, %?i, %?I

プリンタに出力ドット数を送る際に、そのドット数の出力方法を指定しなければなりません。本コード列は、以下に挙げるオプションにおいて、グラフィックデータの出力の指定に使います。

これらのオプション内では、本コード列を何回指定してもかまいません。

○ **汎用プリンタ制御系**

-graphic, -start, -relative, -repeat

○ **ラスタプリンタ制御系**

-Raster_xPos, -Raster_yPos, -Raster_xSize, -Raster_ySize

本コード列は、次に分類するように 6 種類あります。

%?d　? は 1 から 5 までの ASCII コードで、最大 ? 桁までの十進数 ASCII コードで出力することを意味します。

%?D　-graphic, -Raster_xPos, -Raster_yPos, -Raster_xSize および -Raster_ySize でしか使えません。出力するデータがドット数ではなく、バイト数である点を除けば、%?d と同じです。

%?m　? は 1 または 2 で、ドット数を ? バイトのバイナリデータに変換して、ビッグエンディアン[20]で出力することを意味します。

%?M　-graphic, -Raster_xPos, -Raster_yPos, -Raster_xSize および -Raster_ySize でしか使えません。出力するデータがドット数ではなく、バイト数である点を除けば、%?m と同じです。

20) モトローラの CPU が採用している順番で、上位バイトに続き、下位バイトを送ること。

%?i　?は1または2で、ドット数を?バイトのバイナリデータに変換して、リトルエンディアン[21]で出力することを意味します。

%?I　-graphic, -Raster_xPos, -Raster_yPos, -Raster_xSize および -Raster_ySize でしか使えません。出力するデータがドット数ではなく、バイト数である点を除けば、%?i と同じです。

21)インテルの CPU が採用している順番で、下位バイトに続き、上位バイトを送ること。

さらに、次に挙げる2つのオプションに関しては、割り算ファクタを指定することができます。

```
-start, -relative
```

「割り算ファクタ」とは、たとえば 360dpi のプリンタ BJ-130 ではグラフィックリピート指定が 180dpi ベースでしか指定できないというような事態に対処するために考えられた概念です。その正確な定義は以下のようになっています。

定義：　スタートドット位置あるいは相対ドット位置の指定において、出力すべきドット数をある指定数で割り、その商をドット数として出力し、余りは通常のグラフィックデータとしてこのあとに続いて出力する。この指定数を「割り算ファクタ」という。

360dpi 単位で割り出したドット数をすべて 180dpi 単位に変換しなければならないわけですが、単に割り算ファクタ2で割っただけでは余りが出ます。これを無視していてはビットずれのもとですから、それを後続のデータに埋め込んでしまおうというわけです。

書式は、以下に示すとおりです。

```
%?/*d, %?/*m, %?/*i
```

ここで、* の部分に割り算ファクタを書きます。たとえば、BJ-130J では、次のように指定します。

```
-relative=\eU%4/2d%0%0%0
```

この指定は、「ドット単位印字位置相対指定」が「エスケープコード＋U」に続き、最大4桁までの十進数 ASCII コードで相対位置の半分の数値を出力し、3バイトの 0x00 を出力することを意味します。

ちなみに、「エスケープコード＋U」は、180dpi ベースのグラフィックリピート指定です。360dpi の BJ-130J では、空白のドット幅を数えて、その半分を指定してやることが必要なのです。-relative によって、スキップ数の半分を出力してくれますから、リピートすべきドット列 0x00 0x00 0x00 をダミー (%0%0%0) としてあとから送ってやれば、目的を達成できます。

ほかにキャラクタリピート[22]を使う場合、空白文字が幅 48 ドットを持っているなら、次のように指定します。

```
-relative=\eR%3/48d\x20
```

なお、現在、割り算ファクタは複数指定することができません。さらに、割り算ファクタのない指定との混在もできません。

22)たとえば、「エスケープコード＋R *ddd n*」の場合、*ddd* はリピート数を3桁の十進数の ASCII コードで表しており、*n* をリピートすべきキャラクタとすることを示しています。

◆ \x??

1 バイトコードの出力指定です。16 進数で指定します。ちょうど 2 桁分指定してください。ただし、0x00 を出力したい場合は %0 を用いてください。

◆ \???

1 バイトコードの出力指定です。8 進数で指定します。ちょうど 3 桁分指定してください。ただし、0x00 を出力したい場合は %0 を用いてください。

◆ その他のコード

以上に記述したコード列以外にも、以下にリストされた記法が使用できます。

\b	ベル（BEL, 0x07）
\t	水平タブ（HT, 0x09）
\n	改行（LF, 0x0a）
\v	垂直タブ（VT, 0x0b）
\f	改ページ（FF, 0x0c）
\r	復帰（CR, 0x0d）
\d	deselect（DC3, 0x13）
\s	サブコード（SUB, 0x1a）
\e	エスケープコード（ESC, 0x1b）
\\	\コード
%%	% コード
%0	ヌルコード（NUL, 0x00） ヌルコードは \000, \x00 等では指定できません。

1 バイトコード出力指定 (\x?? や \???) だけを使っていると、記述を見ただけでは意味がわかりにくいので、このようなわかりやすい記述のしかたを使うと便利です。

2.5 ················METAFONT のコマンドライン

METAFONT の基本的な実行形式である inimf.x と virmf.x のコマンドライン解釈について簡単に説明しておきます。inimf.x と virmf.x には、通常のプログラムでいう、コマンドラインから指定することで自身の動作を変化させるようなオプションは存在しません。コマンドラインに記述された文字列は、そのまま METAFONT の入力文字として扱われます。

たとえば、コマンドラインから以下のように入力したとしましょう。

```
A> inimf test.mf
```

この記述は、一見するとコマンドラインの引数として test.mf を渡したかのように見えます。しかし、これはただ単に、

```
A> inimf
This is METAFONT, C Version 2.7 (INIMF)
**test.mf
```

とした場合とまったく等価です。コマンドラインからの入力は、そのまま METAFONT の入力として扱われる点に注意してください。

ただし、TEX の場合と同様、“&” を先頭に持つ最初のコマンドライン引数だけは METAFONT でも特別の意味を持ちます。“&” のあとに続く文字列は、inimf.x で作成した、拡張子が .base のファイルを指定します。base ファイルは、TEX にとっての fmt ファイルに相当するファイルです。

“\” (“¥”) は、TEX での場合と同様に METAFONT でもパスの区切りとして認識されませんから、コマンドラインからディレクトリ付きのファイルネームを渡す場合は “\” (“¥”) のかわりに “/” を使用してください。

2.6 ⋯⋯⋯⋯⋯⋯⋯⋯ makefont オプション

makefont.x には、以下に示す各オプションがあります。

- ⟨dpi⟩　　作成フォントの基本 dpi
- ⟨拡大率⟩　　フォントの拡大率
- ⟨フォント名⟩　　作成フォント名
- -mf　　METAFONT ソースファイル検索ディレクトリの追加
- -y　　処理の中断の禁止

⟨dpi⟩ ⟨拡大率⟩ ⟨フォント名⟩ については順番は固定で、省略することはできません。その他のオプションは、これらのオプションの後ろであれば、順不同で、省略も可能です。なお、オプションを指定せずに起動すると、対話形式で makefont.x が起動します。

〈dpi〉〈拡大率〉〈フォント名〉　基本 dpi、拡大率、フォント名の指定

書式：　〈dpi〉〈拡大率〉〈フォント名〉

機能：　作成フォントの基本 dpi、拡大率、フォント名を指定します。

解説：　第 1 オプションとして、作成フォントの基本 dpi を 50–3000 の数値で指定します。プレビューア用のフォント dpi 値は 118 に、プリンタ用のフォントはプリンタの dpi 値にあわせるようにするのが普通です。第 2 オプションとして、作成フォントの拡大率を指定します。拡大率は、以下の表のとおりに指定してください。実際に作成されるフォントサイズは、「dpi×拡大率」の小数点第 1 位が 5 以下のとき、これを切り捨てた値になります。

```
0.5=m05 0.6=m06 0.7=m07 0.8=m08  0.9=m09  1.0=m0   1.095=mh
1.2=m1  1.3=m13 1.44=m2 1.728=m3 2.074=m4 2.488=m5
```

第 3 オプションとして、作成するフォント名を METAFONT のソースファイル名で指定します。拡張子の `.mf` は指定しないでください。指定ソースファイル名の例を以下に示します。

ソースファイル名	作成されるフォント
cmr10	Roman face（ローマン体）
cmti10	*Italic face*（イタリック体）
cmcsc10	SMALL CAPS FACE（小文字サイズの大文字）
cmsl10	*Slanted face*（斜体）
cmss10	Sans serif face（サンセリフ体）
cmtt10	`Typewriter face`（タイプライタ体）

なお、第 1 ～ 第 3 のオプションの順番は固定で、省略することはできません。3 つまとめてひとつのオプションだと考えてください。たとえば、118dpi の cmss10 フォントを拡大率 1.2 で作成する場合、以下のようにします。

```
A> makefont 118 m1 cmss10
```

-mf　METAFONT ソースファイル検索ディレクトリの追加

書式： -mf=〈ディレクトリ名〉

機能： METAFONT ソースファイルの検索ディレクトリを追加します。

解説： METAFONT ソースファイルが標準ディレクトリ以外にある場合に、このオプションで指定します。ディレクトリが環境変数 MFINPUTS で指定されている場合は、このオプションは不要です。

-y　処理の中断の禁止

書式： -y

機能： 処理の中断を行わないようにします。

解説： このオプションがあると処理を中断しません。作成先ディレクトリがない場合は確認することなくディレクトリを作成し、作成フォントがすでに存在する場合は確認することなく作業を終了します。また、METAFONT ソースにエラーがあった場合も、エラーメッセージを出力するだけで作業を続行します。このオプションは、バッチファイル等で連続してフォントを作成するときに指定すると便利です。

CHAPTER 3

Configuration

本章では、フォントマネージャで使用するコンフィギュレーションファイルの文法について解説します。本書の添付ディスクに付属のインストーラによって TEX のインストールを行った場合、フォントマネージャのコンフィギュレーションファイルは、あなたのプリンタおよびフォントファイル環境にあわせて %TEXHOME% 内に myfonts.fm というファイル名で作成されます。しかし、このコンフィギュレーションファイルには、あくまでも最小限度の設定だけしか定義されていませんから、このような既存ファイルの設定に満足できなくなることもあるでしょう。その場合、読者は、自分自身でコンフィギュレーションファイルをつくらなければなりません。本章では、そのような読者のために、フォントマネージャのコンフィギュレーションファイルの書き方について説明します。

3.1…… コンフィギュレーションファイル文法

3.1.1 コンフィギュレーションファイルの文

フォントマネージャのコンフィギュレーションファイルは、基本的に 1 行がひとつの “文” によって構成されています。ここで、“文” とは、フォントマネージャが解析可能な指示命令の集まりのことで、BASIC でいうところの「行」、あるいはバッチファイルでいう「行」の概念に近いものだと思ってもらってもかまいません。

そのようなフォントマネージャのコンフィギュレーションファイルで使われる文は、その働きの違いから大きく 2 種類、正確には 6 種類に分けることができ、それ以外には存在しません。

○ **定義文**

フォントマネージャの働きそのものを決定づける部分です。どのフォントをどのフォントドライバで処理して出力するのかという部分を設定します。

(1) マクロ定義文

複雑で長い文字列に、短くてわかりやすい名前をつけるための文のことです。コンフィギュレーションファイルをわかりやすくするのに効果があります。

(2) ドライバ定義文

どのフォントドライバをどのような設定で処理するのかを定める文のことです。コンフィギュレーションファイルのなかでも特に重要な役割を果たします。

(3) フォント定義文

実際にフォントを作成するために、どのフォントドライバを組み合わせるかを定める文のことです。コンフィギュレーションファイルのなかでも特に重要な役割を果たします。

コンフィギュレーションファイル内では、これらの文を順次並べていくことによって、フォントの登録をプログラミングするわけですが、このとき、各定義文には次のような相互依存関係があります。

- フォント定義文は、いくつかのドライバ定義文を参照する

- フォント定義文やドライバ定義文は、いくつかのマクロ定義文を参照する

このため、文は基本的に、**(1)** → **(2)** → **(3)** の順[1]に並べます。

この点について、もう少し説明してみましょう。

基本的に、フォントマネージャのコンフィギュレーションファイルは、ドライバ定義文によって定義されたフォントドライバの名称 (識別子) を、フォント定義文で列挙することにより、フォントを生成し、また、外部のプログラムが参照できるような名前をつけます。

ここで、「なぜ、フォントネームを使用して記述しないのか」という疑問がわいてくるかもしれません。確かに、“自分の必要なフォントネームを列挙する” だけで、フォントマネージャがフォントドライバの組み合わせや、それらの順序を考えて、自動的にフォントを登録してくれれば理想的でしょう。しかし、それにはちょっと無理があります。なぜなら、融通性を考えると、フォントネームにはあまり多くの情報を盛り込めないからです。

たとえば、`...-tex180` というフォントネームの一部を見ただけではフィルタドライバ `totex.sys` を使っているということだけしかわからず、フィルタドライバに渡したすべての引数を類推することは不可能です。また、キャッシュフィルタの有無をフォントネームから判断することはできません。

逆に、フォントネームにキャッシュフィルタの有無の情報を入れないことで汎用性を確保しています。キャッシュフィルタの有無が、フォントの形を表すフォントネームに影響することは好ましいことではないでしょう。

以上の理由により、コンフィギュレーションファイルの作成者は、引数の書き方も含めたフォントドライバの識別をドライバ定義文で定義し、フォント定義文でどの順番に組み合わせるのかを指示する必要があります。

○ **制御文**

フォントマネージャのコンフィギュレーションファイルに記述された文の流れを制御するための部分です。コンフィギュレーションファイルは、基本的に記述されている順番に文を展開し、処理をしますが、これらの文を用いれば、その処理を制御することができます。

(1) `include` 文

別のファイルに記述された定義文を参照するための文です。コンフィギュレーションファイルをわかりやすくするのに効果があります。

(2) `if` 文

普通のプログラム言語の条件判断文と同じものです。ユーザの環境にあわせた定義をするのに便利です。

(3) `warning` 文、`error` 文

使用者に注意をうながす目的で使う部分です。どちらも通常のプログラム言語の処理中に発生するそれと同じ意味合いを持っていて、`warning` 文は警告を出力させるだけですが、`error` 文は警告を出力したうえ、文の処理も中断させます。

[1] 記述する場合の前後関係は、正確には、定義文が参照しあう順に従って決まってしまいます。

本節の冒頭で述べたように、コンフィギュレーションファイルにはこれら 6 種類以外の文を書くことはできません。

3.1.2 “文” について

前項で述べたように、コンフィギュレーションファイル中の 1 行を “文” といいます。しかし、以下の指定をすることによって、「文は 1 行である」という制限を変更することができます。

◆ 行継続

行末にバックスラッシュ（\）を書くと、その文は次の行につながっていること (行継続) になります。すなわち、「\+ 0x0d + 0x0a」(バックスラッシュおよび改行復帰) は無視されます。

◆ 空行

空行は文とはみなされず、無視されます。

行継続の記号（\）の次に空行がある場合、この空行は無視されてしまいますので、その空行の前の行と次の行は続いているものとみなされます。

◆ コメント

キャラクタ “#” があるとコメントとみなされ、“#” 以下の文章は行末まで無視されます。コメント文が次の行まで続く場合は、行末に “\” をつけても、“\” よりも “#” のほうが強い働きをしますので、“\” は無視されてしまいます。次の行までコメント文を続けたい場合は、次の行の行頭にも “#” をつけてください。

◆ エスケープキャラクタ

“\” は、エスケープキャラクタとして働きます。コンフィギュレーションファイルでは、=, |, [,], $, {, } のほか、アルファベット、数字などが文意によっては特別な意味を持ってきます。それをキャンセルする場合 (特別な意味を持たない単なる文字として使用したい場合、の意味)、これらのキャラクタの前に “\”

を書いてください。たとえば、“=” を特別な意味を持たない単なる文字として使用したい場合には、“\=” と記述します。

また、“\” 自身を通常の文字として認識させたい場合は、“\\” と続けて書いてください。

◆ スペース

スペースは、入れてはならないところ以外では、見やすさのために、

○ いくつ入れてもよい
○ スペースのかわりにタブを使用することができる
○ スペースとタブを混在させて複数書くことができる
○ スペースを入れなくてもよい

ということになっています。

なお、スペースを入れてはならない場合については、以下の説明中、個別に解説します。

3.1.3 “識別子” 定義

識別子は、漢字 (全角文字のすべて)、アルファベットと数字からなる文字列で、マクロ定義文とドライバ定義文によって定義された、いくつもの定義を識別するためのものです。第3.1.1項「コンフィギュレーションファイルの文」(p.100) で、マクロ定義文を「複雑で長い文字列に、短くてわかりやすい名前をつけるための文」のことだと書きましたが、このときに「つけ」られた「短くて使いやすい名前」も、ここでいう “識別子” のひとつです。

識別子では大文字・小文字が使い分けられます。文意あるいは明示的な指定[2]によってマクロネームとドライバネームを区別することができるので、同じ識別子を用いることもできないわけではありませんが、やはり、識別子は別のものにしたほうがいいでしょう。

2)マクロネームの参照は、ネームの前に “$” を置くことで行います。詳しくは、「マクロ定義文」(p.104) を参照。

先に述べたように、識別子に含まれてもよい文字は、漢字とアルファベットと数字ですから、これ以外の文字はすべて識別子のデリミタ (区切り) として機能します。したがって、識別子を記述するときには、文字の間にスペースを入れてはいけません。たとえば識別子 “define” があったとすると、これを “de fine” などと記述することはできません。

3.1.4 文の “フォーム”

第3.1.1項「コンフィギュレーションファイルの文」(p.100) で、コンフィギュ

レーションファイルを構成する文には 6 つの種類があることを簡単に説明しました。本節では、各文について少し詳しく見ていくのと同時に、その文の書き方 (これを「フォーム」と呼びます) を説明しておきます。

その前に、「各フォームに関する説明」の “流れ” を示すとともに、その読み方について、いくつかの注意点を示します。

(1) フォームの書式説明は、フォームの文頭に「○」を付けて行う。

(2) その際、フォームの書式を示す説明部分では、

(a) アルファベットや記号は、実際のフォームの記述どおりである。

(b) カタカナや漢字で記述された部分は、実際のフォームの記述に際して適切な識別子あるいは文字列に置き換えなければならない[3)]。

(3) さらに、フォームの説明のあと、その機能をイメージしてもらうための例を挙げるが、これはフォントマネージャのコンフィギュレーションファイルとして正しいものではない。つまり、あくまでも理解を深めてもらうための材料にすぎない。

(4) 最後にフォントマネージャのコンフィギュレーションファイルとして成り立ちうる実例を示す。

3)具体的にどのような識別子や文字列に置き換えるのかについては、そのつど説明します。

◆ マクロ定義文

フォントマネージャのマクロは、C 言語で使用する、引数をとることも可能な **#define** と同等の働きをします。つまり、「複雑で長い文字列」を「短い識別子」で代用して書いておき、フォントマネージャでこれを解釈するときは、元の長い文字列に置き換えて解釈してほしいというようなときに使います。また、識別子には引数を指定することができますので、“似たような長い文字列” をひとまとめにして定義することができるようになります。

マクロ定義文は、次のフォームをとります。「マクロネーム」および「引数」は識別子であり、「定義体」は文字列です。

○ **define** マクロネーム = 定義体

○ **define** マクロネーム [引数 **1**, 引数 **2**, ..., 引数 **n**] = 定義体

2 つのフォームのうち、上側に示したフォームは、引数をとらないマクロを定義する場合に使用するもので、マクロ定義の最も簡単な利用方法といえます。引数をまったくとらないマクロネームは、この記述方法でしか定義することができません。これに対して下側に示したフォームは、引数をとるマクロを定義する場合に使用し、与えた引数は定義体の中で展開されます。

まず、引数をとらないマクロを定義する場合から詳しく説明します。

マクロ定義文で引数をとらないマクロを定義しておくと、以後、「定義体」の内容を「**$**マクロネーム」で参照することができるようになります。

List 3-1 に、引数をとらないマクロ定義をイメージするための例を示します。

List 3-1● 引数をとらないマクロ定義例

```
 1:  define No = first
 2:  The $No sample.
 3:  define No = second
 4:  The $No sample.
```

この例は、次の展開結果と同等です。

List 3-2● List 3-1 の展開結果

```
 1:  The first sample.
 2:  The second sample.
```

同じマクロネームで新たにマクロ定義文を定義すると、それ以後は新しく定義されたものに置き換わります。List 3-2では、それぞれ **first**, **second** になっていることに注目してください。

それでは、フォントマネージャのコンフィギュレーションファイルの一部として成り立ちうるマクロの使用実例として、List 3-3 を示しましょう。2 行目と 3 行目で引数をとらないマクロを定義し、7, 8, 10 行目でそのマクロを使用しています。

List 3-3● 引数をとらないマクロ定義の実例

```
 1:  #     以下の 2 行が引数をとらないマクロ定義の実例です。
 2:  define freeminsize = 46
 3:  define freeminsize = 48
 4:
 5:  ifdef freeminsize {
 6:    generator MinBitmap = /tex/fontman/bm1.sys \
 7:                              /tex/freefont/mincho$freeminsize.bm1 \
 8:                                 -n free-ビットマップ($freeminsize)-明朝
 9:    dfilter MinBitmapSmooth = /tex/fontman/smooth.sys \
10:                                  -s 256 -p $freeminsize
11:  }
```

この例は、次の展開結果と同等です。

List 3-4● List 3-3 の展開結果

```
 5:  ifdef freeminsize {
 6:    generator MinBitmap = /tex/fontman/bm1.sys \
 7:                              /tex/freefont/mincho48.bm1 \
 8:                                 -n free-ビットマップ(48)-明朝
 9:    dfilter MinBitmapSmooth = /tex/fontman/smooth.sys \
10:                                  -s 256 -p 48
11:  }
```

マクロ **freeminsize** の展開結果 **$freeminsize** が 48 であることに注目してください。

ここで少し寄り道をして、コンフィギュレーションファイルでマクロ参照をする場合のちょっとしたテクニックについて触れておきます。

普通、マクロを参照したときは、そのあとにスペースを入れなければなりません。なぜなら、マクロに引き続いて、さらに識別子になりうるキャラクタ(漢字・

英字・数字) がきた場合、スペースを入れなければ、識別子に続くキャラクタをマクロの識別子の続きとして誤認してしまうことがあるからです。

先の List 3-1 の場合、マクロの識別子に続いていたのが “.”(7 行目), “)”(8 行目) という記号でしたから、これらの文字が同時に識別子のデリミタとしても機能したために、スペースの挿入というような処理は不要でした。

しかし、たとえば、List 3-5 を見てください。

List 3-5● 間違ったマクロの参照例

```
1:  define name = min
2:  file path is B:/TeX/fonts/$name10.pk
```

この例では、name のつもりで書いた識別子が、実際には name10 という識別子として認識されてしまいます。その結果、望んだようにはマクロを展開してくれません。

List 3-6● List 3-5 の展開結果

```
1:  file path is B:/TeX/fonts/$name10.pk
```

では、List 3-7 のように、name と 10 の間にスペースを入れればうまくいくのでしょうか？

List 3-7● 識別子と引数の間にスペースを入れた例

```
1:  define name = min
2:  file path is B:/TeX/fonts/$name 10.pk
```

しかし、実は List 3-7 も希望どおりにはいきません。

List 3-8● List 3-7 の展開結果

```
1:  file path is B:/TeX/fonts/min 10.pk
```

ファイル名 min10.pk が、min 10.pk のように分断されてしまいます。

このような場合は、List 3-9 のように、name と 10 の間に “\” を入れてください。すると、フォントマネージャは、“\” を識別子の終わりとみなします。

List 3-9● マクロ定義テクニック

```
1:  define name = min
2:  file path is B:/TeX/fonts/$name\10.pk
```

これで、ようやく希望どおりのマクロ展開ができるようになります。

List 3-10● List 3-9 の展開結果

```
1:  file path is B:/TeX/fonts/min10.pk
```

なお、マクロ定義の参照は、マクロ定義行を読み込んだ直後に行われるので、次のように define 自体をマクロに置き換えることもできます。

List 3-11● マクロ def を定義する

```
1:  define def = define
```

```
2:  $def A = a
3:  $def B = b
4:  '$A' and '$B' are different characters.
```

List 3-12 ● List 3-11 の展開結果

```
1:  'a' and 'b' are different characters.
```

しかし、このように define 自体をマクロに置き換えるような使い方はコンフィギュレーションファイルが見にくくなるので、通常は使わないようにしてください。

さて次に、引数をとるマクロを定義する場合について説明しましょう。

先にも述べたように、引数をとるマクロを定義した場合、以後、「引数本体」を「定義体」内に展開した内容を、以下に示すフォームで参照することができます。

○ $マクロネーム [引数本体 1, 引数本体 2, ..., 引数本体 n]

引数を定義する場合は、「マクロネーム」に複数の引数を続ける形で行います。引数をとるマクロをイメージするための例を List 3-13 に、その展開結果を List 3-14 に示します。

List 3-13 ● 引数をとるマクロ参照例

```
1:  define X[a,b] = $a are taller than $b.
2:  $X[Elephants,humans]
3:  $X[Giraffes,elephants]
```

List 3-14 ● List 3-13の展開結果

```
1:  Elephants are taller than humans.
2:  Giraffes are taller than elephants.
```

通常、引数をとるマクロは、引数をとらないマクロで代用することができます。たとえば、List 3-15はList 3-13とまったく同等の指定になります。

List 3-15 ● 引数のつかないマクロでの代用例

```
1:  define X = $a are taller than $b.
2:  define a = Elephants
3:  define b = humans
4:  $X
5:  define a = Giraffes
6:  define b = elephants
7:  $X
```

しかし、これは X を定義する時点で a と b が定義されていないので、たまたまうまくいった例です。

たとえば、List 3-16のような場合には、X の定義で a と b が展開されてしまいますから、List 3-17のような展開結果にしかなりません。

List 3-16 ● 代用の失敗例

```
1:  define a = Humans
```

```
2:  define b = dogs
3:  define X = $a are taller than $b.
4:  define a = Horses
5:  define b = humans
6:  $X
7:  define a = Elephants
8:  define b = horses
9:  $X
```

List 3-17● List 3-16の展開結果

```
1:  Humans are taller than dogs.
2:  Humans are taller than dogs.
```

それでは、フォントマネージャのコンフィギュレーションファイルの一部になりうる、引数をとるマクロ定義の実例を List 3-18 に、その展開結果を List 3-19 に示します。

List 3-18● 引数をとるマクロの実例

```
1:  define totex[kind] = totex.sys -j $kind\.tfm -e /dump
2:  filter TotexKMin  = $totex[ZSkmnpaa]
3:  filter TotexKGoth = $totex[ZSkkgpaa]
4:  filter TotexSMin  = $totex[ZSsmnpaa]
5:  filter TotexSGoth = $totex[ZSskgpaa]
```

List 3-19● List 3-18 の展開結果

```
2:  filter TotexKMin  = totex.sys -j ZSkmnpaa.tfm -e /dump
3:  filter TotexKGoth = totex.sys -j ZSkkgpaa.tfm -e /dump
4:  filter TotexSMin  = totex.sys -j ZSsmnpaa.tfm -e /dump
5:  filter TotexSGoth = totex.sys -j ZSskgpaa.tfm -e /dump
```

ところで、フォントマネージャには環境変数を参照するためのマクロがデフォルトで用意されていますので、これについても簡単に触れておくことにしましょう。

このマクロ定義の参照のためのフォームは以下に示すとおりで、このフォームを使用すると、「環境変数名」で指定された環境変数の内容に展開されます。

○ **$env**[環境変数名]

List 3-20 のように書いた場合、環境変数参照マクロを参照している部分が、環境変数 MYNAME の値に展開されるのだとイメージしてください。たとえば、環境変数 MYNAME の値が "Kawamoto" である場合は、List 3-21 のように展開されるイメージです。

List 3-20● 環境変数参照マクロ例

```
1:  My name is $env[MYNAME].
```

List 3-21● 環境変数参照マクロ例

```
1:  My name is Kawamoto.
```

実際のフォントマネージャのコンフィギュレーションファイルにおける使用例としては、List 3-22 に示すようなものがあり、これは環境変数 TEXHOME の内容を展開します。汎用性の高いコンフィギュレーションファイルを記述したい場合には、あなたも同じ記述を使用することがあるかもしれません。

List 3-22● 環境変数参照マクロの実例

```
1: define driver = $env[TEXHOME]/fontman
```

◆ ドライバ定義文

ドライバ定義文は、フォントドライバのファイル名と、そのフォントドライバにどういう引数を渡すのかなどの情報を含めて識別子「ドライバネーム」に定義します。

フォントドライバの種類は FONTMAN Ver.2 までは 3 種類でしたが、FONTMAN Ver.3 から 8 種類に増えました。しかし、コンフィギュレーションファイルのレベルでは 8 種類の分類は必要ないので、次のような 3 分類[4)]で識別しています。

4)FONTMAN Ver.2 で行っていた 3 種類の分類と同じ分類です。

(1) フォントジェネレータ

- ○ ビットマップフォントジェネレータ
- ○ ベクトルフォントジェネレータ

(2) フォントフィルタ

- ○ ビットマップフォントフィルタ
- ○ ベクトルフォントフィルタ
- ○ ビットマップフォントドローア
- ○ ベクトルフォントイクストラクタ

(3) フォントミキサ

- ○ ビットマップフォントミキサ
- ○ ベクトルフォントミキサ

フォントジェネレータ、フォントフィルタ、フォントミキサの概要については『Vol.1 — User's Guide 編』の第 3.2.1 項「概要」(p.50) を参照してください。

フォントマネージャは、フォントジェネレータがターゲットにするフォントがビットマップフォントであるかベクトルフォントであるかを自動判別するので、意識して記述する必要は特にありません。

さて、コンフィギュレーションファイルでドライバ定義文を記述する場合、以下に挙げる 5 種類のフォームが用意されているので、これを使用します。

「ドライバネーム」は識別子で、「ドライバ名とその引数」は文字列です。識別子の参照は、次に説明するフォント定義文で行います。

○ **generator** ドライバネーム = ドライバ名とその引数

フォントジェネレータの定義文です。

○ **filter** ドライバネーム = ドライバ名とその引数

○ **cfilter** ドライバネーム = ドライバ名とその引数

○ **dfilter** ドライバネーム = ドライバ名とその引数

フォントフィルタの定義文です。

フォントマネージャでは、異なるフォントに対し同じフォントフィルタをかけることができます。完全に同じフォントフィルタなら、ひとつのプログラムを共有するとメモリが節約できると考えるのは自然な考え方でしょう。そのため、フォントフィルタ登録に関しては、プログラム領域を共有する手段が用意されています。

しかし、なかには一見共有できるようで、できないフォントフィルタも存在します。プログラム領域を共有できるようにするためには、フォントフィルタの作者は共有されることを意識してプログラミングしなければならないからです。

そこで、フォントマネージャでは、次の3種類の指示方法を用意することでプログラム領域の共有の可否を判断しています。

- filter (ordinary filter)
  他のフィルタとプログラムセクションを共有することがある場合に指定します。
- cfilter (common filter)
  フィルタ同士がプログラムセクションを共有する場合に指定します。
- dfilter (divided filter)
  他のフィルタとプログラムセクションを共有しない場合に指定します。

このうち、使用するフォントフィルタにどの指定が一番適切であるのかは、そのフォントフィルタのドキュメントを参照しなければなりません。

○ **mixer** ドライバネーム = ドライバ名とその引数

フォントミキサの出力にあたる識別子の定義文です。

以上のドライバ定義文を使用したフォントマネージャのコンフィギュレーションファイルの実例として、List 3-23を示します。

List 3-23 ● コンフィギュレーションファイル例

```
 1: define zs = B:/Zsstaff/fonts
 2: define fonts = B:/TeX/fonts/jfonts
 3: define dump = B:/TeX/dump
 4: filter Smooth = D:/smooth.sys
 5: generator Mincho = D:/zs.sys -c -m -s 512 \
 6:                    -1 $zs/mincho.vf1 -2 $zs/mincho.vf2
 7: filter Reverse = D:/reverse.sys
 8: define totex[kind,DPI] = D:/totex.sys -d $DPI \
 9:                          -j $fonts/$kind\10.tfm -e $dump
10: filter TotexMin118 = $totex[min,118]
11: filter TotexMin180 = $totex[min,180]
```

1~3 行目、および 8, 9 行目はマクロ定義文です。最初の 3 行は、ファイルパスをマクロ定義した例です。8, 9 行目は引数付きマクロ定義文の例で、行継続の例でもあります。

ドライバ定義文は 4~7 行目と、10, 11 行目です。4~7 行目は、直接ドライバの引数を書き込んでいる例です。また、10, 11 行目はマクロ定義を利用することで、ほとんど同じ定義の、微妙に違う部分[5] をスマートに定義しています。

5) プレビューアとプリンタドライバが使用するフォントの解像度 (dpi) の違いです。

各フォントドライバへの引数の指定方法は、フォントドライバごとに異なりますので、本書の添付ディスクが構築した TEX システムの場合、`%TEXHOME%\fontman\doc\drivers` 配下にある、それぞれのマニュアルを参照してください。なお、本書の添付ディスクによって構築される TEX システムに含まれているフォントドライバの概略は以下に示すようになっています。

`bdraw.sys`	:	ベクトルフォントをビットマップフォントに変換するフィルタ
`bm1.sys`	:	ビットマップベタファイルフォント用ジェネレータ
`box.sys`	:	フォントのまわりをボックスで囲むフィルタ
`diet.sys`	:	フォントを細らせる（ダイエットする）フィルタ
`fcache.sys`	:	フォントキャッシュフィルタ
`jg.sys`	:	ツァイトのベジェアウトラインフォント用ジェネレータ
`jxl4.sys`	:	アスキーの JXL4 フォーマットファイルフォント用ジェネレータ
`kage.sys`	:	影文字、立体文字、網掛けを生成するフィルタ
`maru.sys`	:	FM TOWNS の丸文字フォント用ジェネレータ
`mixJIS2.sys`	:	アウトラインフォントで第二水準の漢字データがない場合に ROM フォントで代用するためのミキサ
`reverse.sys`	:	白黒反転させるフィルタ
`rombox.sys`	:	本体内蔵の ROM フォント専用のボックスフィルタ
`shiro.sys`	:	白抜きを生成するフィルタ
`smooth.sys`	:	本体内蔵の ROM フォントを拡大縮小してスムージングをかけるフィルタ
`tate.sys`	:	pTEX で必要な縦書きフォントデータを、通常の横書き用フォントから生成するためのフィルタ
`totex.sys`	:	フォントマネージャで生成されたフォントを、TEX に対応させるフィルタ
`vstretch.sys`	:	文字を縦や横から偏平をかけるベクトルフォントフィルタ
`vslant.sys`	:	文字に傾斜をつけるベクトルフォントフィルタ
`zs.sys`	:	ツァイトのリニアアウトラインフォント用ジェネレータ

◆ フォント定義文

フォント定義文は、実際にデバイスドライバなどで使用することができるフォントを作成するために、どのフォントドライバを、どう組み合わせるかを指定するものです。

フォント定義文のフォームは以下に示すとおりです。

○ **font** フォント定義本体

○ **font** フォントエイリアスネーム = フォント定義本体

順番は前後しますが、「フォント定義本体」のみを記述する前者のフォームについては後述することにして、まず、「フォントエイリアスネーム」をあわせて記述する後者のフォームについて説明します。このフォームは、「フォント定義本体」によってできあがったフォントにエイリアスネーム（別名）をつけるためのものです。フォントエイリアスネームには空白記号（スペースとタブ）以外であればどんなキャラクタでも使えますが、その最初は“-”から始まらなければなりません。こうしてエイリアスネームがつけられたフォントは、そのエイリアスネームを p3m ファイルに記述することができます[6]。また、そのコンフィギュレーションファイル全体でフォント定義する場合にも使用できます。

なお、同じフォントエイリアスネームが複数回定義された場合は、あとに定義されたほうが有効になります。

さて、「フォント定義本体」は、次のような再帰的なフォームを持ちます。

各「ドライバネーム」には、すでにドライバ定義文によって定義した識別子を指定しなければなりません。ただし、「ジェネレータドライバネーム」の記述がない (空列) 場合には、ROM フォントを指定したことになります。

6)p3m ファイルとは、フォントマネージャが登録したフォントとデバイスドライバが使用するフォントとの対応を記述したファイルのことです。p3m ファイルへのフォントエイリアスネームの記述例については、List 3-56 (p.131) を参照してください。

○ フォントエイリアスネーム
「フォントエイリアスネーム」は、それ自身でひとつのフォントとなります。

○ ジェネレータドライバネーム
フォントジャネレータを定義する識別子「ジェネレータドライバネーム」は、それ自身でひとつのフォントとなります。

○ フォント定義本体 | フィルタドライバネーム
フォントフィルタを定義する識別子「フィルタドライバネーム」は、親フォントをひとつ指定することで、ひとつのフォントとなります。
フォームについてみると、まず、「フォント定義本体」にフィルタドライバの入力になる親フォントを記述します。次の“|”はパイプ記号で (Human68k の COMMAND.X でいうところのパイプ記号に相当します)、その右側にフォントフィルタを定義する識別子「フィルタドライバネーム」を記述します。

○ { フォント定義本体1, ..., フォント定義本体n } ミキサドライバネーム
フォントミキサは、複数の親フォントを必要とし、ミキサ自身が親フォントをミックスしてひとつのフォントを構成します。

フォームについてみると、左側にフォントミキサの入力となる親フォントを書き並べ、右側にフォントミキサの出力を定義する識別子「ミキサドライバネーム」を記述します。

フォントマネージャのコンフィギュレーションファイルの実例として、簡単で実用的な例を挙げておきます。

List 3-24● 簡単で実用的なコンフィギュレーションファイルの例

```
 1:  define printerDPI = 360
 2:  filter Smooth = D:/smooth.sys
 3:  generator Mincho = D:/zs.sys -c -m -s 256 \
 4:                     -1 B:/Zsfont/mincho.vf1 -2 B:/Zsfont/mincho.vf2
 5:  define totex[kind,DPI] = D:/totex.sys -d $DPI \
 6:                           -j B:/TeX/fonts/$kind\10.tfm -e B:/TeX/dump
 7:  filter TotexMinPreviewer = $totex[min,118]
 8:  filter TotexMinPrinter = $totex[min,$printerDPI]
 9:  filter Reverse = D:/reverse.sys
10:  mixer JisMix = D:/mixJIS2.sys
11:  #
12:  font Mincho|TotexMinPrinter
13:  font Mincho|TotexMinPreviewer
14:  font {Mincho,|Smooth}JisMix|TotexMinPrinter
15:  font -rom-gothic = |Smooth|Reverse
16:  font -rom-gothic-totex118 = -rom-gothic|TotexMinPreviewer
```

最後の 5 行がフォント定義文です。フォントマネージャは、このフォント定義文を読み込むことでフォントの登録順序を決定し、フォントドライバの登録を行います。

(1) zs.sys mincho を登録する (3 行目)。
(2) (1) を totex.sys -d 118 と totex.sys -d 360 に通す (12, 13 行目)。
(3) ROM フォントに smooth.sys を通す (14 行目)。
(4) (1) と (3) のフォントを mixJIS2.sys で混合する (14 行目)。
(5) (4) を totex.sys -d 360 に通す (14 行目)。
(6) (3) を reverse.sys に通し、-rom-gothic というエイリアスネームをつける (15 行目)。
(7) (6) を totex.sys -d 118 に通し、-rom-gothic-totex118 というエイリアスネームをつける (16 行目)。

◆ include 文

include 文は、定義文ではありません。別のファイルに記述された定義文を参照するのに使用する制御文です。そのフォームは、以下に示すとおりです。

○ **include** ファイルパスネーム

「ファイルパスネーム」には、インクルードしたいファイルのパスネームを書きます。なお、ネスティングは深さ 8 まで許されます。include 文が書いてある

と、フォントマネージャはこの行を、指定したファイルの内容で置き換えます。この機能をイメージしてもらうための例を List 3-25 common.fm に示します。

List 3-25● common.fm の内容

```
1:  We are boys.
2:  You are girls.
3:  He he hey. Let's play with us.
```

このとき、List 3-26 は List 3-27 のように展開されます。

List 3-26● include 文使用例

```
1:  Oh my god.
2:  include common.fm
3:  I feel shy.
```

List 3-27● List 3-26の展開結果

```
1:  Oh my god.
2:  We are boys.
3:  You are girls.
4:  He he hey. Let's play with us.
5:  I feel shy.
```

実際のコンフィギュレーションファイルにおける使用例は、ここに挙げるまでもないでしょうから、省略します。

◆ if 文

if 文は定義文ではありません。条件判断を行うための制御文です。if 文には、ifdef 文と ifndef 文、および ifeq 文と ifneq 文があります。

- ifdef 文は、あるマクロネームが定義されているかどうかをチェックし、定義されている場合には、指定された文を実行します。
- ifndef 文は、あるマクロネームが定義されているかどうかをチェックし、定義されていない場合に、指定された文を実行します。
- ifeq 文は、2 つのマクロネームを比較し、その展開結果が等しい場合に、指定された文を実行します。
- ifneq 文は、2 つのマクロネームを比較し、その展開結果が等しくない場合に、指定された文を実行します。

はじめに ifdef 文のフォームを示します。

○ **ifdef マクロネーム 文**

単純 ifdef 文です。「マクロネーム」で示されるマクロが定義されていたとき、指定された「文」が実行されます。ifdef 文が 1 行からなる単純な文である場合に使用します。

ただし、「文」に、もう一度 if 文を書くことはできません。たとえば、「ifdef マクロネーム ifdef ···」という記述はできません。このような文を記述したい場合には、複文の ifdef 文を使ってください。

○ **ifdef マクロネーム {**

```
        ⋮
}
```

else 節をともなわない、複文の ifdef 文です。「マクロネーム」で示されるマクロが定義されていたときに実行する文を、「⋮」の部分に書きます。ここには、複数の文を書くことができるほか、if 文を書くこともできます。ネスティングは、深さ 8 まで許されます。

なお、マクロが定義されていたときに実行する文が 1 行だけで、かつ、if 文でない場合には、単純 ifdef 文を使うことをお勧めします。単純 ifdef 文を使うと、コンフィギュレーションファイルが見やすくなります。

○ **ifdef マクロネーム {**

```
        ⋮
} else {
        ⋮
}
```

else 節をともなう、複文の ifdef 文です。「マクロネームで示されるマクロが定義されているときに実行する文を最初の「⋮」に書き、マクロが定義されていないときに実行する文を else 以降の「⋮」に書きます。ネスティングは、深さ 8 まで許されます。

ifndef 文は、マクロが定義されているかどうかの判断が逆転して、定義されていなかった場合に指定された「文」を実行する以外は ifdef 文と同じですので、ifndef 文の説明は省略します。

次に、ifeq 文について説明します。ifeq 文は、マクロネームを 2 つ必要とし、この 2 つのマクロネームを比較することで分岐を行います。このとき、マクロのどちらかが引数をとっているものであった場合の動作は不確定ですから、ifeq 文での比較対象となるマクロは引数をとってはいけません。

ifeq 文は、以下のフォームをとります。

○ **ifeq マクロネーム マクロネーム 文**

単純 ifeq 文です。2 つの「マクロネーム」の展開結果が等しいときに、指定された「文」を実行します。

ただし、ifeq 文に if 文を書くことはできません。複数の文を書きたい場合には、次に説明する複文の ifeq 文を使ってください。

○ **ifeq マクロネーム マクロネーム {**

```
      ⋮
}
```

else 節をともなわない、複文の ifeq 文です。2 つの「マクロネーム」の展開結果が等しかったときに実行する文を、「 ⋮ 」に書きます。ここには、複数の文を書くことができるほか、if 文を書くこともできます。

ネスティングは、深さ 8 まで許されます。

なお、マクロが定義されていたときに実行する文が 1 行だけで、かつ、if 文でない場合には単純 ifeq 文を使うことをお勧めします。単純 ifeq 文を使うと、コンフィギュレーションファイルが見やすくなります。

○ **ifeq マクロネーム マクロネーム {**

```
    ⋮
} else {
    ⋮
}
```

else 節をともなう複文 ifeq 文です。2 つの「マクロネーム」の展開結果が等しかったときに実行する文を最初の「 ⋮ 」に書き、等しくなかったときに実行する文を else 以降の「 ⋮ 」に書きます。

ネスティングは、深さ 8 まで許されます。

ifneq 文は、マクロネームが等しいかどうかの判断が逆転する以外、ifeq 文と同じですので、ifneq 文の説明は省略します。

◆ warning 文、error 文

warning 文、error 文は、定義文ではありません。warning 文と error 文は、使用者に注意をうながす目的で使います。

文のフォームは以下に示すとおりです。

○ warning 出力文字列

○ error 出力文字列

warning 文は、「出力文字列」を画面に出力したあと、処理自体はそのまま続行します。一方 error 文は、「出力文字列」を画面に出力したあと、それ以後の処理を中止します。

3.1.5 FONTMAN のコンフィギュレーションファイル文法

ここまでに解説してきた、コンフィギュレーションファイルの各種フォームについて、まとめます。

define　マクロ定義フォーム

書式 1：define マクロネーム = 定義体

書式 2：define マクロネーム [引数 1,..., 引数 n] = 定義体

機能：　マクロを定義します。

解説：　引数をとらないマクロの定義例を挙げます。

List 3-28 ● 引数をとらないマクロ定義の実例

```
 1:  #    以下の 2 行が引数をとらないマクロ定義の実例です。
 2:  define freeminsize = 46
 3:  define freeminsize = 48
 4:
 5:  ifdef freeminsize {
 6:    generator MinBitmap = /tex/fontman/bm1.sys \
 7:                /tex/freefont/mincho$freeminsize.bm1 \
 8:                    -n free-ビットマップ($freeminsize)-明朝
 9:    dfilter MinBitmapSmooth = /tex/fontman/smooth.sys \
10:                                  -s 256 -p $freeminsize
11:  }
```

List 3-28 は、List 3-29 のように展開されます。

List 3-29 ● List 3-28 の展開結果

```
 5:  ifdef freeminsize {
 6:    generator MinBitmap = /tex/fontman/bm1.sys \
 7:                             /tex/freefont/mincho48.bm1 \
 8:                                -n free-ビットマップ(48)-明朝
 9:    dfilter MinBitmapSmooth = /tex/fontman/smooth.sys \
10:                                  -s 256 -p 48
11:  }
```

次に、引数をとるマクロの使用例を挙げます。List 3-30 は、List 3-31 に示すように展開されます。

List 3-30 ● 引数をとるマクロの実例

```
1:  define totex[kind] = totex.sys -j $kind\.tfm -e /dump
2:  filter TotexKMin  = $totex[ZSkmnpaa]
3:  filter TotexKGoth = $totex[ZSkkgpaa]
4:  filter TotexSMin  = $totex[ZSsmnpaa]
5:  filter TotexSGoth = $totex[ZSskgpaa]
```

List 3-31 ● List 3-30 の展開結果

```
2:  filter TotexKMin  = totex.sys -j ZSkmnpaa.tfm -e /dump
3:  filter TotexKGoth = totex.sys -j ZSkkgpaa.tfm -e /dump
4:  filter TotexSMin  = totex.sys -j ZSsmnpaa.tfm -e /dump
5:  filter TotexSGoth = totex.sys -j ZSskgpaa.tfm -e /dump
```

env マクロ　環境変数参照マクロフォーム

書式：　$env[環境変数名]

機能：　環境変数の内容を参照します。

解説：　たとえば、環境変数 TEXHOME が A:/TeX であった場合、List 3-32は List 3-33のように展開されます。

List 3-32 ● 環境変数参照マクロの実例

```
1:  define driver = $env[TEXHOME]/fontman
```

List 3-33 ● List 3-32の展開例

```
1:  define driver = A:/TeX/fontman
```

汎用性の高いコンフィギュレーションファイルを作成したい場合には、このマクロを使用するとよいでしょう。

generator フォントジェネレータの定義文フォーム

書式： generator ドライバネーム = ドライバ名とその引数

機能： ビットマップフォントジェネレータあるいはベクトルフォントジェネレータの識別子を定義します。

解説： 「ドライバネーム」は識別子で、「ドライバ名とその引数」は文字列です。識別子の参照はフォント定義文で行います。

コンフィギュレーションファイルの実例を挙げます。

List 3-34● コンフィギュレーションファイルの実例

```
 1: define zs = B:/Zsstaff/fonts
 2: define fonts = B:/TeX/fonts/jfonts
 3: define dump = B:/TeX/dump
 4: filter Smooth = D:/smooth.sys
 5: generator Mincho = D:/zs.sys -c -m -s 512 \
 6:                    -1 $zs/mincho.vf1 -2 $zs/mincho.vf2
 7: filter Reverse = D:/reverse.sys
 8: define totex[kind,DPI] = D:/totex.sys -d $DPI \
 9:                          -j $fonts/$kind\10.tfm -e $dump
10: filter TotexMin118 = $totex[min,118]
11: filter TotexMin180 = $totex[min,180]
```

List 3-34 の 5 行目でフォントジェネレータを、識別子“Mincho”に定義しています。

フォントジェネレータへの引数の指定方法については、フォントジェネレータごとに異なりますので、それぞれのマニュアルを参照してください。

mixer　フォントミキサの定義文フォーム

書式：　mixer ドライバネーム = ドライバ名とその引数

機能：　ビットマップフォントミキサあるいはベクトルフォントミキサを定義します。

解説：　「ドライバネーム」は識別子で、「ドライバ名とその引数」は文字列です。識別子の参照はフォント定義文で行います。

コンフィギュレーションファイルの実例を挙げます。

List 3-35● コンフィギュレーションファイルの実例

```
1:  define driver = $env[TEXHOME]/fontman
2:  mixer JisMix = $driver/mixJIS2.sys
```

List 3-35 の 2 行目でミキサ“JisMix”を定義しています。

フォントミキサへの引数の指定方法については、フォントミキサごとに異なりますので、それぞれのマニュアルを参照してください。

filter, cfilter, dfilter　フォントフィルタの定義文フォーム

書式1：filter ドライバネーム = ドライバ名とその引数

書式2：cfilter ドライバネーム = ドライバ名とその引数

書式3：dfilter ドライバネーム = ドライバ名とその引数

機能：　ビットマップフォントフィルタ、ベクトルフォントフィルタ、ビットマップフォントドローア、ベクトルフォントイクストラクタなどのフォントフィルタを定義します。

○ filter (ordinary filter)

他のフィルタとプログラムセクションを共有することがある場合に指定します。

○ cfilter (common filter)

フィルタどうしがプログラムセクションを共有する場合に指定します。

○ dfilter (divided filter)

他のフィルタとプログラムセクションを共有しない場合に指定します。

解説：　「ドライバネーム」は識別子で、「ドライバ名とその引数」は文字列です。識別子の参照はフォント定義文で行います。

List 3-34 (p.119) の例では、4, 7, 10, 11 行目がフィルタ定義文の例です。それぞれ、スムージングをかけるフィルタ“Smooth”、白黒反転させるフィルタ“Reverse”、118dpi の TEX 情報を付加するためのフィルタ“TotexMin118”、180dpi の TEX 情報を付加するためのフィルタ“TotexMin180”を定義しています。

フォントフィルタへの引数の指定方法については、フォントフィルタごとに異なりますので、それぞれのマニュアルを参照してください。

font　フォント定義文フォーム

書式 1：font フォント定義本体

書式 2：font フォントエイリアスネーム = フォント定義本体

機能：　実際のフォント生成のためのフォントドライバの組み合わせ方を指定します。

解説：　書式 1 では、その行以前に定義済みのフォントやドライバネームを使って新たなフォントを定義します。

書式 2 では、書式 1 と同様に新たなフォントを定義しますが、さらに、そのフォントに名前（エイリアスネーム）をつけます。

「フォント定義本体」は、List 3-36 のような再帰的なフォームを持ちます。

List 3-36 ● フォント定義本体フォーム

```
1:  フォントエイリアスネーム
2:  ジェネレータドライバネーム
3:  フォント定義本体 | フィルタドライバネーム
4:  {フォント定義本体 1,..., フォント定義本体 n} ミキサドライバネーム
```

実用的なコンフィギュレーションファイルを挙げます。

List 3-37 ● コンフィギュレーションファイルの例

```
 1:  define printerDPI = 360
 2:  filter Smooth = D:/smooth.sys
 3:  generator Mincho = D:/zs.sys -c -m -s 256 \
 4:           -1 B:/Zsfont/mincho.vf1 -2 B:/Zsfont/mincho.vf2
 5:  define totex[kind,DPI] = D:/totex.sys -d $DPI \
 6:              -j B:/TeX/fonts/$kind\10.tfm -e B:/TeX/dump
 7:  filter TotexMinPreviewer = $totex[min,118]
 8:  filter TotexMinPrinter = $totex[min,$printerDPI]
 9:  filter Reverse = D:/reverse.sys
10:  mixer JisMix = D:/mixJIS2.sys
11:  #
12:  font Mincho|TotexMinPrinter
13:  font Mincho|TotexMinPreviewer
14:  font {Mincho,|Smooth}JisMix|TotexMinPrinter
15:  font -rom-gothic = |Smooth|Reverse
16:  font -rom-gothic-totex118 = -rom-gothic|TotexMinPreviewer
```

List 3-37 の 12 行目から 14 行目が書式 1 の指定例です。15 行目と 16 行目が書式 2 の指定例です。

include　include 文フォーム

書式：　include ファイルパスネーム

機能：　別のファイルに記述された定義文を参照するのに使用する制御文です。

解説：　「ファイルパスネーム」には、読み込みたいファイルのパスネームを書きます。なお、ネスティングは深さ 8 まで許されます。

include 文は List 3-38 のように使用します。

List 3-38 ● include 文使用例

```
1: define driver = A:/tex/fontman
2: include $driver/configs/drivers.def
3: include $driver/configs/basicfonts.def
4: ifdef useTeX include $driver/configs/texfonts.def
```

ifdef　ifdef 文フォーム

書式 1：ifdef マクロネーム 文

書式 2：ifdef マクロネーム {
　　　⋮
}

書式 3：ifdef マクロネーム {
　　　⋮
} else {
　　　⋮
}

機能：　あるマクロネームが定義されているかどうかをチェックし、定義されている場合には指定された文を実行します。

解説：　以下、順に書式 1、書式 2、書式 3 の使用例を挙げます。

List 3 - 39 ● ifdef 文の使用例　書式 1

```
1:  ifdef useClub define flags = -n zeit-club-ゴシック \
2:                                -c -s $maxFontSizeTwice
```

List 3 - 40 ● ifdef 文の使用例　書式 2

```
1:  ifdef useClub {
2:      define flags = -n zeit-club-ゴシック \
3:                             -c -s $maxFontSizeTwice
4:  }
```

List 3 - 41 ● ifdef 文の使用例　書式 3

```
1:  ifdef useClub {
2:      define flags = -n zeit-club-ゴシック \
3:                             -c -s $maxFontSizeTwice
4:  } else {
5:      define flags = -g -s $maxFontSizeTwice
6:  }
```

useClub というマクロが定義されていたら、フォント名を zeit-club-ゴシックに変えるなどの変更を行います。

ifndef　ifndef 文フォーム

書式 1：ifndef マクロネーム 文
書式 2：ifndef マクロネーム {
　　　　　　　⋮
　　　　}
書式 3：ifndef マクロネーム {
　　　　　　　⋮
　　　　} else {
　　　　　　　⋮
　　　　}

機能：あるマクロネームが定義されているかどうかをチェックし、定義されていない場合には指定された文を実行します。

解説：以下、順に書式 1、書式 2、書式 3 の使用例を挙げます。

List 3-42● ifndef 文の使用例　書式 1

```
1:  ifndef min2font define ZSMIN = {$ZSMIN,$ROMMIN}JisMix
```

List 3-43● ifndef 文の使用例　書式 2

```
1:  ifndef min2font {
2:              define ZSMIN = {$ZSMIN,$ROMMIN}JisMix
3:  }
```

List 3-44● ifndef 文の使用例　書式 3

```
1:  ifndef min2font {
2:              define ZSMIN = {$ZSMIN,$ROMMIN}JisMix
3:  } else {
4:              define ZSMIN = {$ZSMIN,$BMMIN}JisMix
5:  }
```

マクロ min2font が定義されてなければ、第二水準の明朝体の使用をあきらめ、フォントミキサを使って ROM フォントで代用するなどします。

ifeq　ifeq 文フォーム

書式 1：ifeq マクロネーム マクロネーム 文

書式 2：ifeq マクロネーム マクロネーム {
　　　⋮
}

書式 3：ifeq マクロネーム マクロネーム {
　　　⋮
} else {
　　　⋮
}

機能：　2 つのマクロを比較し、展開結果が等しい場合に指定された文を実行します。

解説：　以下、順に書式 1、書式 2、書式 3 の使用例を挙げます。

List 3-45 ● ifeq 文の使用例　書式 1

```
1:  define yes = YES
2:  define driver = $env[TEXHOME]
3:  ifeq useBitmap yes include $driver/configs/bitmap.fm
```

List 3-46 ● ifeq 文の使用例　書式 2

```
1:  define yes = YES
2:  define driver = $env[TEXHOME]
3:  ifeq useBitmap yes {
4:          include $driver/configs/bitmap.fm
5:  }
```

List 3-47 ● ifeq 文の使用例　書式 3

```
1:  define yes = YES
2:  define driver = $env[TEXHOME]
3:  ifeq useBitmap yes {
4:          include $driver/configs/bitmap.fm
5:  } else {
6:          include $driver/configs/bitmap2.fm
7:  }
```

マクロ useBitmap の展開結果が YES であれば、bitmap.fm をインクルードするなどします。

ifneq　ifneq 文フォーム

書式 1：ifneq マクロネーム マクロネーム 文

書式 2：ifneq マクロネーム マクロネーム {
⋮
}

書式 3：ifneq マクロネーム マクロネーム {
⋮
} else {
⋮
}

機能：　2 つのマクロを比較し、展開結果が等しい場合に指定された文を実行します。

解説：　以下、順に書式 1、書式 2、書式 3 の使用例を挙げます。

List 3-48 ● ifneq 文の使用例　書式 1

```
1:  define default = ZMIN
2:  define driver = $env[TEXHOME]
3:  ifneq zeitfontname default error フォント \
4:                                 $zeitfontname には未対応です
```

List 3-49 ● ifneq 文の使用例　書式 2

```
1:  define default = ZMIN
2:  define driver = $env[TEXHOME]
3:  ifneq zeitfontname default {
4:              error フォント $zeitfontname には未対応です
5:  }
```

List 3-50 ● ifneq 文の使用例　書式 3

```
1:  define default = ZMIN
2:  define driver = $env[TEXHOME]
3:  ifneq zeitfontname default {
4:              error フォント $zeitfontname には未対応です
5:  } else {
6:              error フォント $zeitfontname に対応しています
7:  }
```

マクロ zeitfontname の展開結果が ZMIN でなければ、未対応メッセージを出力して異常終了するなどします。

warning　warning 文フォーム

書式：　warning 出力文字列

機能：　「出力文字列」を画面に出力したあと、そのまま続行します。

解説：　warning 文をデバッグに使用する例を挙げます。

List 3-51 ● warning 文使用例

```
1:  ifndef useTeX warning TeX 用のフォントが生成されません
```

マクロ useTeX が未定義の場合、ウォーニングメッセージを出力します。

error　error 文フォーム

書式：　error 出力文字列

機能：　「出力文字列」を画面に出力したあと、それ以後の実行を中止します。

解説：　error 文を使用する例を挙げます。

List 3-52 ● error 文使用例

```
1:  ifndef useZeit error Zeit 用のフォントが生成されません
```

マクロ useZeit が未定義の場合、エラーメッセージを出力して異常終了します。

3.2 ······························ p3m ファイル

X680x0 版 TEX に導入されたフォントマネージャシステムと、TEX ドライバ群との間ではフォントネームの変換を行う必要があります。本節では、その変換規則を書いたファイルである p3m ファイルのフォーマットについて説明します。

3.2.1 p3m ファイル概要

TEX システムでは、英語フォントはデータを pk ファイルに収めて利用します。しかし、日本語フォントはデータ量が膨大なため、英語フォントと同じように pk ファイルに収めると、ディスクスペースを大量に消費します。

そこで、X680x0 版 TEX では、日本語フォントにかぎってフォントマネージャでリアルタイムにフォントを生成する形をとっています。

『Vol.1 — User's Guide 編』の第 3.2.1 項「概要」(p.50) で述べたように、フォントマネージャは TEX で使用されるだけではなく、各種アプリケーションで利用することができる汎用のフォントマネージメントシステムです。そのフォントマネージャの汎用性を高めるため、TEX をはじめとするフォントマネージャを利用する各種アプリケーションとフォントマネージャとの間のインターフェースについては詳細な取り決めがあります。

たとえば、TEX システムにおけるフォントマネージャを考えた場合、TEX とフォントマネージャとの間でフォントネームの変換を行う必要があります。そのためのファイルが p3m ファイルです。普通に TEX をインストールすると、TEX の HOME ディレクトリ内に `preview.p3m` と `print.p3m` という名前のファイルができています。`preview.p3m` がプレビューアを実行する際に参照される p3m ファイルで、`print.p3m` がプリンタドライバを実行する際に参照される p3m ファイルです。

通常、使用する場合は、これらの p3m ファイルのフォーマットについて詳しく知っている必要はありません。しかし、新たなフォントを追加する場合等、通常の使用範囲を越える使い方をする場合には、p3m ファイルのフォーマットと、p3m ファイルの書き換えについての知識が必要になります。

本節では、`print.p3m` を例にとりながら、p3m ファイルの書き方について説明し、フォントマネージャで通常定義されているフォント名についても説明します。

3.2.2 p3m ファイルフォーマット

本項では、print.p3m のサンプルを例にとりながら、p3m ファイルのフォーマットについて説明します。

List 3-53● print.p3m の例

```
 1: #
 2: # sample of print.p3m
 3: #
 4: 118     min          -tex-高速-明朝
 5: 118     goth         -tex-高速-ゴシック
 6: 118     tmin         -tex-高速-縦-明朝
 7: 118     tgoth        -tex-高速-縦-ゴシック
 8:         180/min10    -tex-高速-明朝
 9:         min          -tex-明朝
10:         goth         -tex-ゴシック
11:         tmin         -tex-縦-明朝
12:         tgoth        -tex-縦-ゴシック
```

List 3-53を見てください。1〜3 行目が「コメント」、8 行目が「例外定義」、残りの行が「一般定義」となっています。

「例外定義」も「一般定義」も、

○ 省略可能な基本 dpi 値
○ 省略不可能な pk ファイル名
○ 省略不可能なフォントエイリアスネーム(あるいはフォントネーム)

の 3 つのフィールドから構成されています。

「基本 dpi 値」は、ドライバの dpi 値を特定する場合に記述します。これを明記すると、ドライバの dpi 値が明記した dpi 値に一致する場合だけ、この行の定義が有効になります。省略された場合には、ドライバの dpi 値がなんであっても、その行の指定が採用されます。

「例外定義」は、ひとつの pk ファイル[1]ごとに、どのフォントエイリアス[2]を使用するかを指定するためのものです。基本 dpi 値が明記されていれば、ドライバの dpi 値と一致した場合にかぎって採用されます。

「一般定義」は、デバイスの dpi 値が基本 dpi 値と一致する場合にかぎって、そのフォントのあらゆるポイントの、あらゆるスケールについて同一のフォントエイリアス[3] を採用することを表します。基本 dpi 値が省略されていれば、ドライバの dpi 値を選びません。

TEX のデバイスドライバが、指定された pk ファイルに対してどのフォントエイリアスを採用するのか、これらの定義の解釈順序を明確に書いておきます。

(1) まず、ファイルの最初から順に例外定義を探す。見つかったら、残りは無視される。

(2) 例外定義がなければ、ファイルの最初から順に一般定義を探す。見つかったら、残りは無視される。

1)上の例では、180 dpi の min10.pk。

2)同じく、上の例では、-tex-高速-明朝。

3)フォントネームでも同じです。

(3) 一般定義もなければ、従来の FONTMAN Ver.1 と同様に、*.pk ファイルを探す。

たとえば、List 3-54 の定義は 2 行目が優先ですが、List 3-55 のように指定すると 1 行目が優先されます。

List 3-54● 2 行目が優先

```
1:        min             -tex-明朝
2:        180/min10       -tex-高速-明朝
```

List 3-55● 1 行目が優先

```
1:        180     min     -tex-高速-明朝
2:        118     min     -tex-高速-明朝
```

3.2.3 フォントマネージャで定義されるフォントエイリアス

通常、p3m ファイルには、フォントネームよりフォントエイリアスを書くことを心がけてください。フォントネームはすべてのフォントセットを区別するための名前であるのに対し、フォントエイリアスは機能としての名前なので、これを使用すれば fontman.x が自動的に[4]その機能に一番ふさわしいフォントを割り当てるようになっているからです。

本書添付のコンフィギュレーションファイル myfonts.fm で登録されている標準的なエイリアスは、次のとおりです。

4)正確には fontman.fm 等のフォントマネージャコンフィギュレーションファイル内で指定します。

List 3-56● 標準フォントエイリアス

```
1:  -通常                   通常フォント
2:  -強調                   -通常 を強調したもの
3:  -明朝                   明朝体
4:  -ゴシック               ゴシック体
5:  -tex-高速-明朝           TeX (主に プレビューア) 用高速明朝体
6:  -tex-高速-ゴシック       TeX (主に プレビューア) 用高速ゴシック体
7:  -tex-明朝                TeX (主に プリンタドライバ) 用明朝体
8:  -tex-ゴシック            TeX (主に プリンタドライバ) 用ゴシック体
```

たとえば、「-tex-明朝」の場合、普通、ツァイト社のアウトラインフォント[5]を意味します。しかし、アウトラインフォントがなければ、ROM フォントを拡大・縮小したフォント[6] を使います。また、「強調」という指定にも 3 種類[7]のバリエーションがあります。これらはフォントマネージャを登録する際にユーザの環境、あるいは趣味にあわせて決めることができます。

p3m ファイルにエイリアスネームを指定しておけば、たとえフォントがバージョンアップして、よりきれいなフォントに置き換えることになっても、変更するのはフォントマネージャのコンフィギュレーションだけですむことになり、保守がたいへん楽になります。

さて、TeX のデバイスドライバと関係のあるフォントエイリアス[8]について説明しましょう。

5)たとえば、zeit-club-明朝-diet-tex。

6)たとえば、sharp-rom-平滑-tex。

7)白抜き、ボックス、反転の 3 種類。

8)fontman.fm で登録されるフォントエイリアスです。

List 3-56で -tex が行頭についているものが TEX 関連の必須フォントです。プリンタの種類には関係なく、またプレビューアだろうとプリンタドライバだろうと一貫して使うことができます。しかし、他のフォントは、適宜[9] 登録しなければ使えません。たとえば、丸文字フォントを登録すると、List 3-57 のフォントが使えるようになります。

9)fontman.fm で明示的に登録します。

List 3-57● 丸文字フォントエイリアス

```
1:  -tex-丸                    TeX（主にプリンタドライバ）用丸文字体
```

また、ツァイトの毛筆体や教科書体を登録すると、List 3-58 のフォントが使えるようになります。

List 3-58● 毛筆体・教科書体フォントエイリアス

```
1:  -tex-毛筆                  TeX（主にプリンタドライバ）用毛筆体
2:  -tex-教科書                TeX（主にプリンタドライバ）用教科書体
```

次に、縦書きフォントを登録すると、List 3-59 のフォントが登録されます。

List 3-59● 縦書きフォントエイリアス

```
1:  -tex-高速-縦-明朝          TeX（主にプレビューア）用高速縦書き明朝体
2:  -tex-高速-縦-ゴシック      TeX（主にプレビューア）用高速縦書きゴシック体
3:  -tex-縦-明朝               TeX（主にプリンタドライバ）用縦書き明朝体
4:  -tex-縦-ゴシック           TeX（主にプリンタドライバ）用縦書きゴシック体
```

もちろん、丸文字があれば、さらに List 3-60 のフォントが、ツァイトの毛筆体や教科書体があれば、List 3-61 のフォントがそれぞれ登録されます。

List 3-60● 丸文字縦書きフォントエイリアス

```
1:  -tex-縦-丸                 TeX（主に previewer）用縦書き丸文字体
```

List 3-61● 毛筆体・教科書体縦書きフォントエイリアス

```
1:  -tex-縦-毛筆               TeX（主に printer driver）用縦書き毛筆体
2:  -tex-縦-教科書             TeX（主に printer driver）用縦書き教科書体
```

毛筆体や教科書体としてはツァイト社から発売されているフォントが 1 種類しかありませんが、ツァイトの書体倶楽部にはどんどん新しいフォントが追加されていますし、他社のアウトラインフォントが X680x0 でも使えるようになるかもしれません。仮にフォントの実体を将来変更することになったとしても、フォントエイリアスを同じものにすることによって、修正箇所を最小限に抑えることができます。

なお、現在、どんなフォントエイリアスが登録されているかを確認するときは、fontman -a としてください。また、どんなフォントネームが登録されているかを見るときは、fontman -n とします。さらに、p3m ファイルに書くことができるフォントネームをすべて表示したければ、fontman -l とします。詳しくは、第 2.2 節「FONTMAN オプション」を参照してください。

CHAPTER 4

TEX Fonts

TEX では、METAFONT が作成する、さまざまな英数字および記号のフォントを利用することができます。本書がサポートしているのは、そのなかでも D. E. Knuth 博士によって作成され、数あるフォントのなかでも標準となっている CM (Computer Modern) フォントです。本章では、その多彩な CM フォントを利用するために用意された LaTeX のコントロール・シーケンスの一覧、および METAFONT のフォントテーブルの一覧をお届けします。

4.1 ……………………………… 特殊記号

本節では、LATEX で特殊記号を出力するためのコントロール・シーケンスを紹介します。

なお、数学記号の出力を定義したコントロール・シーケンスの多くは数式モードでしか利用できませんから、注意してください。

4.1.1　主に文章中で利用する記号

パラグラフモードおよび LR モードで使用することができる、あるいは動作モードを問わずに使用することができるというコントロール・シーケンスを紹介しましょう。

◆ 特殊な働きをする記号

LATEX には特殊な働きをするためにソース中にそのまま記述することができない記号がありますが、そのような記号を出力するためのコントロール・シーケンスを下表に示します。

これらのコントロール・シーケンスは LATEX のソースであるかぎり、モードを問わずに利用できるといってよいでしょう。

Table 4 - 1● 特殊な働きをする記号

出力	ソース	出力	ソース	出力	ソース
\	{\tt\symbol{'134}}	{	\{	}	\}
^	{\tt\symbol{'136}}	_	_	%	\%
~	{\tt\symbol{'176}}	$	\$	&	\&
				#	\#

◆ アクセント記号

パラグラフモードおよび LR モードで利用することができるコントロール・シーケンスです。

Table 4-2● アクセント記号

出力	ソース	出力	ソース	出力	ソース	出力	ソース
ò	\`{o}	ó	\'{o}	ö	\"{o}	ȯ	\.{o}
õ	\~{o}	ō	\={o}	ŏ	\u{o}	ǒ	\v{o}
ô	\^{o}	o͡o	\t{oo}	ő	\H{o}	ọ	\d{o}
o̧	\c{o}	o̲	\b{o}				

◆ そのほかの記号

以下に示す Table 4-3 中のコントロール・シーケンスは、すべてのモードで利用することができます。

Table 4-3● そのほかの記号

出力	ソース	出力	ソース
†	\dag	‡	\ddag
§	\S	¶	\P
©	\copyright	£	\pounds

これに対して、以下に示す Table 4-4 はパラグラフモードおよび LR モードで利用することができるコントロール・シーケンスです。

Table 4-4● 発音記号など

出力	ソース	出力	ソース	出力	ソース	出力	ソース
æ	\ae	Æ	\AE	œ	\oe	Œ	\OE
å	\aa	Å	\AA	ł	\l	Ł	\L
ø	\o	Ø	\O	ß	\ss		
¡	!`	¿	?`				

4.1.2 数学記号

本項で示すコントロール・シーケンスは数式モードでしか扱うことができませんから、注意しなければなりません。

◆ ギリシャ文字

Table 4 - 5 ● ギリシャ文字

出力	ソース	出力	ソース	出力	ソース	出力	ソース
A	A	α	\alpha	Ξ	\Xi	ξ	\xi
B	B	β	\beta	O	O	o	o
Γ	\Gamma	γ	\gamma	Π	\Pi	π	\pi
Δ	\Delta	δ	\delta			ϖ	\varpi
E	E	ϵ	\epsilon	P	P	ρ	\rho
		ε	\varepsilon			ϱ	\varrho
Z	Z	ζ	\zeta	Σ	\Sigma	σ	\sigma
H	H	η	\eta			ς	\varsigma
Θ	\Theta	θ	\theta	T	T	τ	\tau
		ϑ	\vartheta	Υ	\Upsilon	υ	\upsilon
I	I	ι	\iota	Φ	\Phi	ϕ	\phi
K	K	κ	\kappa			φ	\varphi
Λ	\Lambda	λ	\lambda	X	X	χ	\chi
M	M	μ	\mu	Ψ	\Psi	ψ	\psi
N	N	ν	\nu	Ω	\Omega	ω	\omega

◆ 矢印

Table 4 - 6 ● 矢印

出力	ソース	出力	ソース	出力	ソース
$\leftarrow$	\leftarrow	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\rightarrow$	\rightarrow	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons	$\leadsto$	\leadsto		

◆ 二項演算記号

Table 4 - 7 ● 二項演算記号

出力	ソース	出力	ソース	出力	ソース	出力	ソース
$\pm$	\pm	$\cap$	\cap	$\diamond$	\diamond	$\oplus$	\oplus
$\mp$	\mp	$\cup$	\cup	$\bigtriangleup$	\bigtriangleup	$\ominus$	\ominus
$\times$	\times	$\uplus$	\uplus	$\bigtriangledown$	\bigtriangledown	$\otimes$	\otimes
$\div$	\div	$\sqcap$	\sqcap	$\triangleleft$	\triangleleft	$\oslash$	\oslash
$\ast$	\ast	$\sqcup$	\sqcup	$\triangleright$	\triangleright	$\odot$	\odot
$\star$	\star	$\vee$	\vee	$\lhd$	\lhd	$\bigcirc$	\bigcirc
$\circ$	\circ	$\wedge$	\wedge	$\rhd$	\rhd	$\dagger$	\dagger
$\bullet$	\bullet	$\setminus$	\setminus	$\unlhd$	\unlhd	$\ddagger$	\ddagger
$\cdot$	\cdot	$\wr$	\wr	$\unrhd$	\unrhd	$\amalg$	\amalg

◆ 関係記号

Table 4 - 8 ● 関係記号

出力	ソース	出力	ソース	出力	ソース	出力	ソース
$\leq$	\leq	$\geq$	\geq	$\equiv$	\equiv	$\models$	\models
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim	$\perp$	\perp
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq	$\mid$	\mid
$\ll$	\ll	$\gg$	\gg	$\asymp$	\asymp	$\parallel$	\parallel
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx	$\bowtie$	\bowtie
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong	$\Join$	\Join
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\neq$	\neq	$\smile$	\smile
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\doteq$	\doteq	$\frown$	\frown
$\in$	\in	$\ni$	\ni	$\propto$	\propto		
$\vdash$	\vdash	$\dashv$	\dashv				

◆ そのほかの数学記号

Table 4 - 9 ● そのほかの数学記号

出力	ソース	出力	ソース	出力	ソース	出力	ソース
$\aleph$	\aleph	$\prime$	\prime	$\exists$	\exists	$\Diamond$	\Diamond
$\hbar$	\hbar	$\emptyset$	\emptyset	$\neg$	\neg	$\triangle$	\triangle
$\imath$	\imath	∇	\nabla	$\flat$	\flat	$\clubsuit$	\clubsuit
$\jmath$	\jmath	$\surd$	\surd	$\natural$	\natural	$\diamondsuit$	\diamondsuit
ℓ	\ell	$\top$	\top	$\sharp$	\sharp	$\heartsuit$	\heartsuit
$\wp$	\wp	$\bot$	\bot	$\backslash$	\backslash	$\spadesuit$	\spadesuit
$\Re$	\Re	$\|$	\\|	∂	\partial	$\ldots$	\ldots
$\Im$	\Im	$\angle$	\angle	∞	\infty	$\cdots$	\cdots
$\mho$	\mho	$\forall$	\forall	$\Box$	\Box	$\vdots$	\vdots
						$\ddots$	\ddots

◆ 大きさが変化する数学記号

インライン数式環境で使用した場合とディスプレイ数式環境で使用した場合とでは出力される大きさが変化する数学記号です。以下の表において、「出力 (I)」はインライン数式環境での出力例を、「出力 (D)」はディスプレイ数式環境での出力例を示します。

Table 4 - 10 ● 大きさが変化する数学記号

出力 (I)	出力 (D)	ソース	出力 (I)	出力 (D)	ソース
$\sum$	$\displaystyle\sum$	\sum	$\bigcap$	$\displaystyle\bigcap$	\bigcap
$\prod$	$\displaystyle\prod$	\prog	$\bigcup$	$\displaystyle\bigcup$	\bigcup
$\coprod$	$\displaystyle\coprod$	\coprod	$\bigsqcup$	$\displaystyle\bigsqcup$	\bigsqcup
$\int$	$\displaystyle\int$	\int	$\bigvee$	$\displaystyle\bigvee$	\bigvee
$\oint$	$\displaystyle\oint$	\oint	$\bigwedge$	$\displaystyle\bigwedge$	\bigwedge
$\bigodot$	$\displaystyle\bigodot$	\bigodot	$\bigotimes$	$\displaystyle\bigotimes$	\bigotimes
$\bigoplus$	$\displaystyle\bigoplus$	\bigoplus	$\biguplus$	$\displaystyle\biguplus$	\biguplus

◆ かっこなどの記号

Table 4 - 11 ● かっこなどの記号

<table>
<tr><th>出力</th><th>ソース</th><th>出力</th><th>ソース</th></tr>
<tr><td>(</td><td>(</td><td>)</td><td>)</td></tr>
<tr><td>[</td><td>[</td><td>]</td><td>]</td></tr>
<tr><td>{</td><td>\{</td><td>}</td><td>\}</td></tr>
<tr><td>⌊</td><td>\lfloor</td><td>⌋</td><td>\rfloor</td></tr>
<tr><td>⌈</td><td>\lceil</td><td>⌉</td><td>\rceil</td></tr>
<tr><td>⟨</td><td>\langle</td><td>⟩</td><td>\rangle</td></tr>
<tr><td>/</td><td>/</td><td>\</td><td>\backslash</td></tr>
<tr><td>|</td><td>|</td><td>‖</td><td>\|</td></tr>
</table>

◆ 関数などの記号

Table 4 - 12 ● 関数などの記号

出力	ソース	出力	ソース	出力	ソース
arccos	\arccos	dim	\dim	log	\log
arcsin	\arcsin	exp	\exp	max	\max
arctan	\arctan	gcd	\gcd	min	\min
arg	\arg	hom	\hom	Pr	\Pr
cos	\cos	inf	\inf	sec	\sec
cosh	\cosh	ker	\ker	sin	\sin
cot	\cot	lg	\lg	sinh	\sinh
coth	\coth	lim	\lim	sup	\sup
csc	\csc	lim inf	\liminf	tan	\tan
deg	\deg	lim sup	\limsup	tanh	\tanh
det	\det	ln	\ln		

◆ アクセント記号

Table 4 - 13 ● アクセント記号

出力	ソース	出力	ソース	出力	ソース	出力	ソース
$\hat{a}$	\hat{a}	$\acute{a}$	\acute{a}	$\bar{a}$	\bar{a}	$\dot{a}$	\dot{a}
$\check{a}$	\check{a}	$\grave{a}$	\grave{a}	$\vec{a}$	\vec{a}	$\ddot{a}$	\ddot{a}
$\breve{a}$	\breve{a}	$\tilde{a}$	\tilde{a}				

4.2 ························ ポイント別文字サイズ比較

以下に示すのは、ポイント別の文字サイズ比較です。

どのコントロール・シーケンスがどのポイントのフォントを出力するかについては、『Vol.1 — User's Guide 編』の第 4.5.1 項「文字の大きさを変更する」(p.137) を参照してください[1]。

1)11, 14, 20, 25pt の英字が、やや太い線でかかれているように見えると思います。これは、これらのサイズ専用にデザインされた METAFONT のフォントがないために、10pt のフォントを単純に拡大して使用しているためです。この件についての詳細は、本書の『Vol.1 — User's Guide 編』、p.139 のコラムに示しています。

5pt : cmr5 and min5

SAMPLE sample みほん 見本

6pt : cmr6 and min6

SAMPLE sample みほん 見本

7pt : cmr7 and min7

SAMPLE sample みほん 見本

8pt : cmr8 and min8

SAMPLE sample みほん 見本

9pt : cmr9 and min9

SAMPLE sample みほん 見本

10pt : cmr10 and min10

SAMPLE sample みほん 見本

11pt : cmr10 \magstephalf and min10 \magstephalf

SAMPLE sample みほん 見本

12pt : cmr12 and min10 \magstep1

SAMPLE sample みほん 見本

14pt : cmr10 \magstep2 and min10 \magstep2

SAMPLE sample みほん 見本

17pt : cmr17 and min10 \magstep3

SAMPLE sample みほん 見本

20pt : cmr10 \magstep4 and min10 \magstep4

SAMPLE sample みほん 見本

25pt : cmr10 \magstep5 and min10 \magstep5

SAMPLE sample

みほん 見本

4.3 .... Font Tables

本節には、本書の添付ディスクで構築した METAFONT システムで作成可能なフォントのうち、主に 10 ポイントのフォントにかぎってフォントテーブルを掲載しています。

4.3.1 掲載フォント一覧

掲載順序は以下のようになっています。

○ **Computer Modern Font Family**

CM フォント・ファミリーのフォントです。フォントを、用途別およびシェイプ別に大きく分類し、この分類に従って並べています。ただし、

(1) ここでの分類は本章筆者のきわめて個人的な趣味によるものであること
(2) 分類の際に同じ系列にあると判断したフォントはできるだけ見開きページにまとめるように配慮した結果、掲載の順番は必ずしも使用頻度を反映していないこと

この 2 点には注意してください。

- Roman Fonts Variation

`cmr10` : CM roman p.146
`cmb10` : CM bold roman p.146
`cmbx10` : CM bold extended roman p.147
`cmfib8` : CM roman fibonacci font p.147

- *Slanted Roman Fonts Variation*

`cmsl10` : CM slanted roman p.148
`cmbxsl10` : CM bold extended slanted roman p.148

- Caps and Small Caps Fonts

`cmcsc10` : CM caps and small caps p.149

- *Italic Fonts Variation*

`cmti10` : CM text italic p.150
`cmbxti10` : CM bold extended text italic p.150
`cmu10` : CM unslanted italic p.151

- **Typewriter Fonts Variation**

cmtt10	: CM typewriter text	p.152
cmvtt10	: CM variable-width typewriter	p.152
cmsltt10	: CM slanted typewriter	p.153
cmitt10	: CM italic typewriter	p.153
cmtcsc10	: CM typewriter caps and small caps	p.154
cmtex10	: CM TEX extended ASCII characters	p.154

- Sans Serif Fonts Variation

cmssq8	: CM sans serif quotation style	p.155
cmssqi8	: CM sans serif quotation style slanted	p.155
cmss10	: CM sans serif	p.156
cmssdc10	: CM sans serif demibold condensed	p.156
cmssbx10	: CM sans serif bold extended	p.157
cmssi10	: CM slanted sans serif	p.157

- Etc. Fonts

cmff10	: CM funny roman	p.158
cmfi10	: CM funny italic	p.158
cmdunh10	: CM dunhill roman	p.159
cminch	: CM inch-high sans serif bold extended caps and digits	p.160

- Math Fonts

cmex10	: CM math extension	p.163
cmmi10	: CM math italic	p.164
cmmib10	: CM math italic bold	p.165
cmsy10	: CM math symbols	p.166
cmbsy10	: CM bold math symbols	p.167

○ **LATEX Extended Font Family**

簡単な図形を描画するために用意されたLATEX用のフォントの一覧です。LATEX用のフォントは、フォントファイルの頭に「l」をつけることが決まりになっています[1)]。

1) この約束事ができる前につくられた一部のLATEXソースファイルでは、「l」のつかないファイル名でフォントを参照している場合もあります。そのようなソースを処理する場合には、ソース自体を修正するか、またはフォントをコピーしてリネームするか、いずれかの手段で対処してください。

lasy10	: LATEX symbols	p.168
lasyb10	: LATEX bold symbols	p.169
line10	: LATEX line	p.168
linew10	: LATEX wide line	p.169
lcircle10	: LATEX circle	p.170
lcirclew10	: LATEX wide circle	p.172

○ METAFONT's logo, and so on...

D. E. Knuth 博士が『TEXbook』および『METAFONT book』のために作成した `manfont` など、特殊記号フォントの一覧です。

logo10	: METAFONT logo	p.174
logobf10	: boldface METAFONT logo	p.174
logosl10	: slanted METAFONT logo	p.174
manfnt	: Special font for the TEX and METAFONT manuals	p.175
ascgrp	: ASCII special font family, special graphic font ...	p.176

4.3.2 フォントテーブルの参照にあたって

さて、テーブルの見方について、簡単に説明しておきましょう。Table 4-14 (p.143) を参照しながら読んでください。

フォントテーブルの左上にはフォントのファイル名 (の拡張子を除いた部分) を、傍注欄にはフォントの正式名称、および本書の添付ディスクによって構築した TEX システムに用意されているデザインサイズ[2] を、それぞれ示しました。Table 4-14 の場合、ファイル名 cmr10 の CM Roman フォントのフォントテーブルであり、5, 6, 7, 8, 9, 10, 12, 17pt というデザインサイズの同名フォント[3] が用意されていることになります。

テキスト記述に用いる (あるいは用いることもできる) 類のフォントについては、フォントテーブルの左側にフォントの情報を示しています[4]。

○ 1em

当該フォントに定められている 1em の幅[5] を意味します。

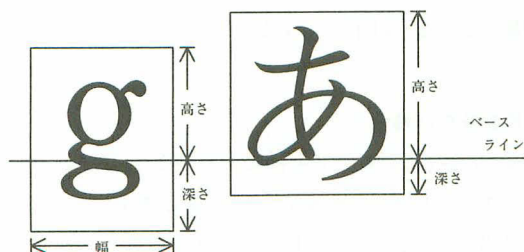
○ 1ex

当該フォントに定められている 1ex の高さ[6] を意味します。

○ height

当該フォント中、0 ··· 9, A ··· Z, a ··· z の ASCII コードと等しい TEX の内部コードを持つ文字[7] のうち、最も高い文字の高さを意味します。

TEX の世界において「文字の高さ」は、“ベースラインより上の部分の高さ”を意味します。ベースラインとは、言葉のとおり基準となる線のことです。たとえば「ABCDEABCDE」というとき、各文字はその下端を架空の直線に揃えているように見えます。この架空の線を「ベースライン」いい、TEX はこの線にあわせて文字を並べることで組版を実現します (Fig. 4-1)。

Fig. 4-1● フォントのベースライン、および高さと深さ

文字の「深さ」は、一見英数字や記号にだけしか設定されていないかのようにも見えますが、日本語フォントにも設定されています。

2)フォントが何ポイントのサイズ用につくられたものであるかを示す値のことをいいます。

3)cmr5,cmr6,...,cmr12, cmr17 があるということを意味します。

4)情報を得るための (ソースレベルの) 方法については、%TEXHOME%\macros\fontbl.tex を参照してください。

5)1em については『Vol.1 — User's Guide 編』の p.140 のコラムを参照してください。

6)1ex については『Vol.1 — User's Guide 編』の p.140 のコラムを参照してください。

7)TEX の内部コードは ASCII コードをもとにしていますから、当該フォントファイルが 0···9, A···Z, a···z を含むなら、それらの文字自身を示す TEX の内部コードは ASCII コードと一致します。また、それらの文字を含まない場合でも、コード同士の対応関係は変わりませんから、それらの文字を含んでいるフォントテーブルと比較することで、たやすくコード同士の対応関係を把握することができるでしょう。

Table 4-14 ● フォントテーブルのサンプル

cmr10[8]

1em = 10.00002pt

1ex = 4.30554pt

height = 6.94444pt

depth = 1.94444pt

slant = 0.0pt

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

8)CM roman (5, 6, 7, 8, 9, 10, 12, 17pt)

したがって、たとえば「文字“y”の高さ」という場合、「AByCD」としてみると明らかな、“ベースラインより下にはみだした部分”はそれに含まれません。この“はみだし部分”、つまりベースラインより下の部分の高さをTEXの世界では「文字の深さ」いいます。

なお、フォントテーブル中、左見出しのイタリック体で書かれた数字（Table 4-14 でいえば“*'??x*”という数字列）のベースラインと、要素となっている文字のベースラインとを揃えてあります。ベースラインの位置確認をしたい場合は参考にしてください。

○ depth

当該フォント中、0 ⋯ 9, A ⋯ Z, a ⋯ z の ASCII コードと等しい TEX の内部コードを持つ文字のうち、最も深い文字の深さを意味します。

○ slant

当該フォントの傾きを意味します[9]。

9)実際にはアクセント記号を付ける位置の調整のために TEX 内部で利用される値です。

また、plain TEX の `\char`、あるいは LATEX の `\symbol` のように、フォントの文字コード（TEX の内部コード）を指定しなければならないコントロール・シーケンスを使用する場合などは、以下に示すような手順を踏んでフォントの文字コードを得てください。

○ **8 進数で文字コードを得たい場合**

仮に cmr10 における「Z」の文字コードを得たいものとしましょう。

(1) まず、cmr10 のフォントテーブルから「Z」を探します。

(2) テーブルの左見出しにイタリック体で記述された数字に注目し、文字コードの上位桁を得ます。

「Z」の場合、*'13x*[10] がこれにあたります。

(3) テーブルの上見出しにイタリック体で記述された数字に注目し、文字コードの下位桁を得ます。
「Z」の場合、*'2* がこれにあたります。

(4) 「Z」の文字コードが *'132* であることがわかりました[11]。

○ **16 進数で文字コードを得たい場合**

仮に cmr10 における「R」および「Z」の文字コードを得たいものとしましょう。

(1) まず、cmr10 のフォントテーブルから「R」および「Z」を探します。

(2) テーブルの右見出しにタイプライタ体で記述された数字に注目し、文字コードの上位桁を得ます。
「R」および「Z」とも、`"5x`[12] がこれにあたります。

(3) 下位桁の取得には若干の注意を要します。今求めた上位桁は、フォントテーブルでいう 2 列ずつをひとつの値で示していました[13]。もし、文字コードを得たい文字がこの 2 列のうちの上側の列にある場合、テーブルの上見出しにイタリック体で記述された数字が下位桁を意味します。これに対して、文字コードを得たい文字が 2 列のうちの下側の列にある場合、テーブルの下見出しにタイプライタ体で記述された数字が下位桁を意味します。
「R」は、上位桁 `"5x` が表すフォントテーブル上の 2 列のうち上の列にあるので、*'2* が下位桁にあたります。これに対し、「Z」は、上位桁 `"5x` が表す 2 列のうち下の列にあるので、`"A` が下位桁にあたります。

(4) 「R」の文字コードが `"52`、「Z」の文字コードが `"5A` であるとわかりました。

4.3.3 Font Tables

それでは、次ページから始まる多種多彩なフォント群を活用してください。

10) TeX では数字の前に「 ' 」をつけることで、8 進数であることを表します。

11) 上位桁で x で表されていた部分に、下位桁が収まったわけです。

12) TeX では数字の前に「 " 」をつけることで、16 進数であることを表します。

13) 16 進数で文字コードを表す場合、上位桁はフォントテーブル上の 2 列ずつをひとつの値で示します。16 進数の "10" が 8 進数の "20" であることを考えれば、これは不思議でもなんでもありません。

14)CM roman (5, 6, 7, 8, 9, 10, 12, 17pt)

cmr10[14)]

1em = *10.00002pt*

1ex = *4.30554pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

15)CM bold roman (10pt)

cmb10[15)]

1em = *10.00002pt*

1ex = *4.44444pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	**Γ**	**Δ**	**Θ**	**Λ**	**Ξ**	**Π**	**Σ**	**Υ**	"0x
'01x	**Φ**	**Ψ**	**Ω**	**ff**	**fi**	**fl**	**ffi**	**ffl**	
'02x	**ı**	**ȷ**	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	**ß**	**æ**	**œ**	**ø**	**Æ**	**Œ**	**Ø**	
'04x	-	**!**	”	**#**	**$**	**%**	**&**	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	"3x
'07x	**8**	**9**	:	;	¡	=	¿	?	
'10x	**@**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	"4x
'11x	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	
'12x	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	"5x
'13x	**X**	**Y**	**Z**	[	“	]	ˆ	˙	
'14x	‘	**a**	**b**	**c**	**d**	**e**	**f**	**g**	"6x
'15x	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	
'16x	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	"7x
'17x	**x**	**y**	**z**	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmbx10[16)]

1em = *11.49994pt*

1ex = *4.44444pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

16)CM bold extended roman (5, 6, 7, 8, 9, 10 12pt)

cmfib8[17)]

1em = *10.49991pt*

1ex = *4.0pt*

height = *6.47223pt*

depth = *1.52777pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

17)CM roman fibonacci font (8pt)

18)CM slanted roman (8, 9, 10, 12pt)

cmsl10[18)]

1em = *10.00002pt*

1ex = *4.30554pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.16667pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

19)CM bold extended slanted roman (10pt)

cmbxsl10[19)]

1em = *11.49994pt*

1ex = *4.44444pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.16667pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmcsc10[20)]

1em = *11.05545pt*

1ex = *4.30554pt*

height = *6.83331pt*

depth = *1.94444pt*

slant = *0.0pt*

	'0	*'1*	*'2*	*'3*	*'4*	*'5*	*'6*	*'7*	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	↑	↓	'	¡	¿	
'02x	I	J	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	SS	Æ	Œ	Ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	<	=	>	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	A	B	C	D	E	F	G	"6x
'15x	H	I	J	K	L	M	N	O	
'16x	P	Q	R	S	T	U	V	W	"7x
'17x	X	Y	Z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

20)CM caps and small caps (10pt)

21)CM text italic (7, 8, 9, 10, 12pt)

cmti10[21)]

1em = *10.22217pt*

1ex = *4.30554pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.25pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	£	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

22)CM bold extended text italic (10pt)

cmbxti10[22)]

1em = *11.8221pt*

1ex = *4.44444pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.25pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	£	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmu10[23)]

1em = *11.11111pt*

1ex = *4.30554pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	"	#	£	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	"	]	ˆ	˙	
'14x	'	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

23)CM unslanted italic (10pt)

24)CM typewriter text (8, 9, 10, 12pt)

cmtt10[24)]

1em = *10.4999pt*

1ex = *4.30554pt*

height = *6.11111pt*

depth = *2.22223pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	↑	↓	'	¡	¿	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	␣	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	<	=	>	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	\	]	^	_	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	{	|	}	~	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

25)CM variable-width typewriter (10pt)

cmvtt10[25)]

1em = *10.4999pt*

1ex = *4.30554pt*

height = *6.11111pt*

depth = *2.22223pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmsltt10[26]

1em = *10.4999pt*

1ex = *4.30554pt*

height = *6.11111pt*

depth = *2.22223pt*

slant = *0.16667pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	↑	↓	'	¡	¿	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	␣	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	<	=	>	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	\	]	^	_	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	{	|	}	~	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

26)CM slanted typewriter (10pt)

cmitt10[27]

1em = *10.4999pt*

1ex = *4.30554pt*

height = *6.11111pt*

depth = *2.22223pt*

slant = *0.25pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	↑	↓	'	¡	¿	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	␣	!	"	#	£	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	<	=	>	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	\	]	^	_	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	{	|	}	~	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

27)CM italic typewriter (10pt)

28) CM typewriter caps and small caps (10pt)

cmtcsc10[28)]

1em = *10.4999pt*

1ex = *4.30554pt*

height = *6.11111pt*

depth = *1.38887pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	↑	↓	'	¡	¿	
'02x	I	J	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	SS	Æ	Œ	Ø	Æ	Œ	Ø	
'04x	␣	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	<	=	>	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	\	]	^	_	
'14x	'	A	B	C	D	E	F	G	"6x
'15x	H	I	J	K	L	M	N	O	
'16x	P	Q	R	S	T	U	V	W	"7x
'17x	X	Y	Z	{	|	}	~	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

29) CM TEX extended ASCII characters (8, 9, 10pt)

cmtex10[29)]

1em = *10.4999pt*

1ex = *4.30554pt*

height = *6.11111pt*

depth = *2.22223pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	·	↓	α	β	∧	¬	∈	π	"0x
'01x	λ	γ	δ	↑	±	⊕	∞	∂	
'02x	⊂	⊃	∩	∪	∀	∃	⊗	⇆	"1x
'03x	←	→	≠	◇	≤	≥	≡	∨	
'04x		!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	<	=	>	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	\	]	^	_	
'14x	'	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	{	|	}	~	∫	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmssq8[30]

1em = *10.00003pt*

1ex = *4.16667pt*

height = *5.55556pt*

depth = *1.11111pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	"	]	ˆ	˙	
'14x	'	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

30)CM sans serif quotation style (8pt)

cmssqi8[31]

1em = *10.00003pt*

1ex = *4.16667pt*

height = *5.55556pt*

depth = *1.11111pt*

slant = *0.21255pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	*Γ*	*Δ*	*Θ*	*Λ*	*Ξ*	*Π*	*Σ*	*Υ*	"0x
'01x	*Φ*	*Ψ*	*Ω*	*ff*	*fi*	*fl*	*ffi*	*ffl*	
'02x	*ı*	*ȷ*	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	*ß*	*æ*	*œ*	*ø*	*Æ*	*Œ*	*Ø*	
'04x	-	*!*	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	"3x
'07x	*8*	*9*	:	;	*¡*	=	*¿*	*?*	
'10x	@	*A*	*B*	*C*	*D*	*E*	*F*	*G*	"4x
'11x	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	
'12x	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	"5x
'13x	*X*	*Y*	*Z*	[	"	]	ˆ	˙	
'14x	'	*a*	*b*	*c*	*d*	*e*	*f*	*g*	"6x
'15x	*h*	*i*	*j*	*k*	*l*	*m*	*n*	*o*	
'16x	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	"7x
'17x	*x*	*y*	*z*	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

31)CM sans serif quotation style slanted (8pt)

32)CM sans serif (8, 9, 10, 12, 17pt)

cmss10[32)]

1em = *10.00002pt*

1ex = *4.44444pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	ϒ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	"	]	^	˙	
'14x	'	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	~	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

33)CM sans serif demi-bold condensed (10pt)

cmssdc10[33)]

1em = *9.49986pt*

1ex = *4.72223pt*

height = *6.94444pt*

depth = *1.66666pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	ϒ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	"	]	^	˙	
'14x	'	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	~	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmssbx10[34)]

1em = *11.00006pt*

1ex = *4.58333pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	ϒ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

34)CM sans serif bold extended (10pt)

cmssi10[35)]

1em = *10.00002pt*

1ex = *4.44444pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.21255pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	*Γ*	*Δ*	*Θ*	*Λ*	*Ξ*	*Π*	*Σ*	*ϒ*	"0x
'01x	*Φ*	*Ψ*	*Ω*	*ff*	*fi*	*fl*	*ffi*	*ffl*	
'02x	*ı*	*ȷ*	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	*ß*	*æ*	*œ*	*ø*	*Æ*	*Œ*	*Ø*	
'04x	-	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	"3x
'07x	*8*	*9*	:	;	¡	=	¿	?	
'10x	@	*A*	*B*	*C*	*D*	*E*	*F*	*G*	"4x
'11x	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	
'12x	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	"5x
'13x	*X*	*Y*	*Z*	[	“	]	ˆ	˙	
'14x	‘	*a*	*b*	*c*	*d*	*e*	*f*	*g*	"6x
'15x	*h*	*i*	*j*	*k*	*l*	*m*	*n*	*o*	
'16x	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	"7x
'17x	*x*	*y*	*z*	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

35)CM slanted sans serif (8, 9, 10, 12, 17pt)

36)CM funny roman (10pt)

cmff10[36)]

1em = 8.2222pt

1ex = 5.27776pt

height = 6.25pt

depth = 2.77776pt

slant = -0.1pt

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	˗	!	"	#	$	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	"	]	ˆ	˙	
'14x	'	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

37)CM funny italic (10pt)

cmfi10[37)]

1em = 10.72214pt

1ex = 5.27776pt

height = 6.25pt

depth = 2.77776pt

slant = 0.1pt

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	˗	!	"	#	£	%	&	'	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	"	]	ˆ	˙	
'14x	'	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmdunh10[38)]

1em = *10.00002pt*

1ex = *4.30554pt*

height = *9.72223pt*

depth = *1.94444pt*

slant = *0.0pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
'02x	ı	ȷ	`	´	ˇ	˘	¯	˚	"1x
'03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
'04x	-	!	”	#	$	%	&	’	"2x
'05x	(	)	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	¡	=	¿	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[	“	]	ˆ	˙	
'14x	‘	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	–	—	˝	˜	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

38)CM dunhill roman (10pt)

39) CM inch-high sans serif bold extended caps and digits (10pt)

cminch[39] - (1)

	'0	'1
'06x	0	1
'07x	8	9
'10x		A
'11x	H	I
'12x	P	Q
'13x	X	Y
	"8	"9

1em = *114.47557pt*

1ex = *47.69814pt*

height = *72.26999pt*

depth = *10.98503pt*

slant = *0.0pt*

cminch - (2)

'2	'3	'4	'5
2	3	4	5
B	C	D	E
J	K	L	M
R	S	T	U
Z			
"A	"B	"C	"D

cminch - (3)

'6	'7	
6	7	"3x
F	G	"4x
N	O	
V	W	"5x
"E	"F	

cmex10[40]

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	(	)	[	]	⌊	⌋	⌈	⌉	"0x
'01x	{	}	⟨	⟩	∣	∥	/	\	
'02x	(	)	(	)	[	]	⌊	⌋	"1x
'03x	⌈	⌉	{	}	⟨	⟩	/	\	
'04x	(	)	[	]	⌊	⌋	⌈	⌉	"2x
'05x	{	}	⟨	⟩	/	\	/	\	
'06x	⎛	⎞	⎡	⎤	⎣	⎦	⎢	⎥	"3x
'07x	⎧	⎫	⎩	⎭	⎨	⎬	⎪	⏐	
'10x	⎝	⎠	⎜	⎟	⟨	⟩	⨆	⨆	"4x
'11x	∮	∮	⨀	⨀	⨁	⨁	⨂	⨂	
'12x	∑	∏	∫	⋃	⋂	⨄	⋀	⋁	"5x
'13x	∑	∏	∫	⋃	⋂	⨄	⋀	⋁	
'14x	∐	∐	ˆ	ˆ	ˆ	˜	˜	˜	"6x
'15x	[	]	⌊	⌋	⌈	⌉	{	}	
'16x	√	√	√	√	⎷	∣	⌜	∥	"7x
'17x	↑	↓	◜	◝	◟	◞	⇑	⇓	
	"8	"9	"A	"B	"C	"D	"E	"F	

40)CM math extension (10pt)

41)CM math italic (5, 6, 7, 8, 9, 10, 12pt)

cmmi10[41)]

1em = *10.00002pt*

1ex = *4.30554pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.25pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	$\mathit{\Gamma}$	$\mathit{\Delta}$	$\mathit{\Theta}$	$\mathit{\Lambda}$	$\mathit{\Xi}$	$\mathit{\Pi}$	$\mathit{\Sigma}$	$\mathit{\Upsilon}$	"0x
'01x	$\mathit{\Phi}$	$\mathit{\Psi}$	$\mathit{\Omega}$	α	β	γ	δ	ϵ	
'02x	ζ	η	θ	ι	κ	λ	μ	ν	"1x
'03x	ξ	π	ρ	σ	τ	υ	ϕ	χ	
'04x	ψ	ω	ε	ϑ	ϖ	ϱ	ς	φ	"2x
'05x	$\leftharpoonup$	$\leftharpoondown$	$\rightharpoonup$	$\rightharpoondown$	‘	’	$\triangleright$	$\triangleleft$	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	.	,	$<$	/	$>$	$\star$	
'10x	∂	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	$\flat$	$\natural$	$\sharp$	$\smile$	$\frown$	
'14x	ℓ	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	$\imath$	$\jmath$	$\wp$	$\vec{}$	$\frown$	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmmib10[42]

1em = *11.49994pt*

1ex = *4.44444pt*

height = *6.94444pt*

depth = *1.94444pt*

slant = *0.25pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
'01x	Φ	Ψ	Ω	α	β	γ	δ	ϵ	
'02x	ζ	η	θ	ι	κ	λ	μ	ν	"1x
'03x	ξ	π	ρ	σ	τ	υ	ϕ	χ	
'04x	ψ	ω	ε	ϑ	ϖ	ϱ	ς	φ	"2x
'05x	$\leftharpoonup$	$\leftharpoondown$	$\rightharpoonup$	$\rightharpoondown$	$\lhook$	$\rhook$	$\triangleright$	$\triangleleft$	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	.	,	<	/	>	$\star$	
'10x	∂	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	$\flat$	$\natural$	$\sharp$	$\smile$	$\frown$	
'14x	ℓ	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	$\imath$	$\jmath$	$\wp$	$\vec{}$	$\frown$	
	"8	"9	"A	"B	"C	"D	"E	"F	

42)CM math italic bold (10pt)

43)CM math symbols (5, 6, 7, 8, 9, 10pt)

cmsy10[43)]

1em = *10.00002pt*

1ex = *4.30554pt*

height = *7.5pt*

depth = *9.6pt*

slant = *0.25pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	$-$	$\cdot$	$\times$	$\ast$	$\div$	$\diamond$	$\pm$	$\mp$	"0x
'01x	$\oplus$	$\ominus$	$\otimes$	$\oslash$	$\odot$	$\bigcirc$	$\circ$	$\bullet$	
'02x	$\asymp$	$\equiv$	$\subseteq$	$\supseteq$	$\leq$	$\geq$	$\preceq$	$\succeq$	"1x
'03x	$\sim$	$\approx$	$\subset$	$\supset$	$\ll$	$\gg$	$\prec$	$\succ$	
'04x	$\leftarrow$	$\rightarrow$	$\uparrow$	$\downarrow$	$\leftrightarrow$	$\nearrow$	$\searrow$	$\simeq$	"2x
'05x	$\Leftarrow$	$\Rightarrow$	$\Uparrow$	$\Downarrow$	$\Leftrightarrow$	$\nwarrow$	$\swarrow$	$\propto$	
'06x	$\prime$	∞	$\in$	$\ni$	$\triangle$	$\triangledown$	$/$	$\mapstochar$	"3x
'07x	$\forall$	$\exists$	$\neg$	$\emptyset$	$\Re$	$\Im$	$\top$	$\bot$	
'10x	$\aleph$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	$\mathcal{E}$	$\mathcal{F}$	$\mathcal{G}$	"4x
'11x	$\mathcal{H}$	$\mathcal{I}$	$\mathcal{J}$	$\mathcal{K}$	$\mathcal{L}$	$\mathcal{M}$	$\mathcal{N}$	$\mathcal{O}$	
'12x	$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{T}$	$\mathcal{U}$	$\mathcal{V}$	$\mathcal{W}$	"5x
'13x	$\mathcal{X}$	$\mathcal{Y}$	$\mathcal{Z}$	$\cup$	$\cap$	$\uplus$	$\wedge$	$\vee$	
'14x	$\vdash$	$\dashv$	$\lfloor$	$\rfloor$	$\lceil$	$\rceil$	$\{$	$\}$	"6x
'15x	$\langle$	$\rangle$	$\vert$	$\Vert$	$\updownarrow$	$\Updownarrow$	$\backslash$	$\wr$	
'16x	$\surd$	$\amalg$	∇	$\smallint$	$\sqcup$	$\sqcap$	$\sqsubseteq$	$\sqsupseteq$	"7x
'17x	$\S$	$\dagger$	$\ddagger$	$\P$	$\clubsuit$	$\diamondsuit$	$\heartsuit$	$\spadesuit$	
	"8	"9	"A	"B	"C	"D	"E	"F	

cmbsy10[44]

1em = *11.49994pt*

1ex = *4.44444pt*

height = *7.5pt*

depth = *9.4pt*

slant = *0.25pt*

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	−	·	×	*	÷	⋄	±	∓	"0x
'01x	⊕	⊖	⊗	⊘	⊙	◯	∘	•	
'02x	≍	≡	⊆	⊇	≤	≥	⪯	⪰	"1x
'03x	∼	≈	⊂	⊃	≪	≫	≺	≻	
'04x	←	→	↑	↓	↔	↗	↘	≃	"2x
'05x	⇐	⇒	⇑	⇓	⇔	↖	↙	∝	
'06x	′	∞	∈	∋	△	▽	/	ꜜ	"3x
'07x	∀	∃	¬	∅	ℜ	ℑ	⊤	⊥	
'10x	ℵ	𝒜	ℬ	𝒞	𝒟	ℰ	ℱ	𝒢	"4x
'11x	ℋ	ℐ	𝒥	𝒦	ℒ	ℳ	𝒩	𝒪	
'12x	𝒫	𝒬	ℛ	𝒮	𝒯	𝒰	𝒱	𝒲	"5x
'13x	𝒳	𝒴	𝒵	∪	∩	⊎	∧	∨	
'14x	⊢	⊣	⌊	⌋	⌈	⌉	{	}	"6x
'15x	⟨	⟩	$\vert$	$\Vert$	↕	⇕	\	≀	
'16x	√	∐	∇	∫	⊔	⊓	⊑	⊒	"7x
'17x	§	†	‡	¶	♣	♢	♡	♠	
	"8	"9	"A	"B	"C	"D	"E	"F	

44)CM bold math symbols (10pt)

45) LaTEX symbols (5, 6, 7, 8, 9, 10pt)

lasy10[45]

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x		◁	⊴	▷	⊵				"0x
'01x									
'04x									"2x
'05x	<	>	^	v					
'06x	℧	⋈	□	◇					"3x
'07x			∼	⇝	⊏	⊐			
	"8	"9	"A	"B	"C	"D	"E	"F	

46) LaTEX line (10pt)

line10[46]

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x									"0x
'01x									
'02x									"1x
'03x									
'04x									"2x
'05x									
'06x									"3x
'07x									
'10x									"4x
'11x									
'12x									"5x
'13x									
'14x									"6x
'15x									
'16x									"7x
'17x									
	"8	"9	"A	"B	"C	"D	"E	"F	

lasyb10[47]

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x		◁	⊴	▷	⊵				"0x
'01x									
'04x									"2x
'05x	‹	›	^	˅					
'06x	℧	⋈	□	◇					"3x
'07x			∼	⇝	⊏	⊐			
	"8	"9	"A	"B	"C	"D	"E	"F	

47) LaTeX bold symbols (10pt)

linew10[48]

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	/	/	/	/	/	/			"0x
'01x	/	▲	/	▶	/	▼	◀	▼	
'02x	/	/	◀	/	/	◀		▲	"1x
'03x	/	▲	/	◀	/	▶	◀		
'04x	/	/	/	/		/			"2x
'05x	/	▲	▼	▲	/	▶		▶	
'06x		▼		▼			▲	◀	"3x
'07x		▲	▶			▲	▼	▼	
'10x	\	\	\	\	\	\			"4x
'11x	\	▼	\	▶	\	▲	▼	▲	
'12x	\	\	▲	\	\	◀		▼	"5x
'13x	\	▼	\		\	▶	◀		
'14x	\	\	\	\		\			"6x
'15x	\	▼	▲	▼	\			▶	
'16x		▲		▲				◀	"7x
'17x		◀	▶			▼	▲		
	"8	"9	"A	"B	"C	"D	"E	"F	

48) LaTeX wide line (10pt)

49) LATEX circle (10pt)

lcircle10[49] - (1)

	'0	'1	'2	'3
'00x				
'01x				
'02x				
'03x				
'04x				
'05x				
'14x				
'15x				
'16x				
'17x				
	"8	"9	"A	"B

51) METAFONT logo (10pt)

logo10[51)]

1em = 7.99997pt
1ex = 0.0pt
height = 6.0pt
depth = 0.0pt
slant = 0.0pt

	'0	'1	'2	'3	'4	'5	'6	'7	
'10x		A				E	F		"4x
'11x						M	N	O	
'12x					T				"5x
'13x									
	"8	"9	"A	"B	"C	"D	"E	"F	

52) boldface METAFONT logo (10pt)

logobf10[52)]

1em = 9.39996pt
1ex = 0.0pt
height = 6.0pt
depth = 0.0pt
slant = 0.0pt

	'0	'1	'2	'3	'4	'5	'6	'7	
'10x		**A**				**E**	**F**		"4x
'11x						**M**	**N**	**O**	
'12x					**T**				"5x
'13x									
	"8	"9	"A	"B	"C	"D	"E	"F	

53) slanted METAFONT logo (10pt)

logosl10[53)]

1em = 7.99997pt
1ex = 0.0pt
height = 6.0pt
depth = 0.0pt
slant = 0.25pt

	'0	'1	'2	'3	'4	'5	'6	'7	
'10x		*A*				*E*	*F*		"4x
'11x						*M*	*N*	*O*	
'12x					*T*				"5x
'13x									
	"8	"9	"A	"B	"C	"D	"E	"F	

manfnt[54)]

54)Special font for the TeX and METAFONT manuals (10pt)

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x							A	A	"0x
'01x	A	A	A	A	A	A	A	A	
'02x	A								"1x
'03x		S	O	I					
'04x									"2x
'05x	M	E	T	A	F	O	N	M	
'06x	E	T	A	F	O	N	▲	▼	"3x
'07x	*M*	*E*	*T*	*A*	*F*	*O*	*N*		
'10x		A	M	E	T	E	F	A	"4x
'11x	F	O	N	M	E	M	N	O	
'12x	.		*		T	T	A	F	"5x
'13x					*M*	*E*	*T*	*A*	
'14x						*F*	*O*	*N*	"6x
'15x	M	E	T	A	F	O	N	M	
'16x	E	T	A	F	O	N		N	"7x
'17x	▶								
	"8	"9	"A	"B	"C	"D	"E	"F	

55) ASCII special font family, special graphic font (10pt)

ascgrp[55)]

	'0	'1	'2	'3	'4	'5	'6	'7	
'04x									"2x
'05x									
'06x	▶	◀	▲	▼					"3x
'07x									
'10x									"4x
'11x									
'14x									"6x
'15x									
	"8	"9	"A	"B	"C	"D	"E	"F	

X68k Programming Series #3 TEX

INDEX

記号

& 24, 94
-a 32
-buf 46
-CRLF 79, 91
-dpi 43, 61
-dump 73
-EveryRaster 87
-extraCRLF 80, 91
-FF 81, 91
-fontmanMaxSize 52, 71
-GRAM 45, 64
-graphic 82, 91
-height 47, 67
-highReso 45, 46, 47
-i=$ 〈数値〉 36
-i= 〈フォントネーム〉 37
-i 34
-info 41, 58
-init 78, 91
-l 31
-landscape 65
-layout 52, 71
-mag 44, 62
-mf 97
-MH 87
-MSBisLeft 75
-MSBisUpper 74
-n 33
-paperCol 50
-penCol 50
-pinBytes 76
-pinHeights 76
-pk2fontman 10, 15, 51, 70
-pk3fontman 11, 16, 51, 70
-prBufSize 77
-r 38
-Raster 86
-Raster_xPos 89, 91
-Raster_xSize 90, 91
-Raster_yPos 89, 91
-Raster_ySize 90, 91
-RasterOutPutOrder 88
-relative 84, 91
-remark 40, 57
-repeat 85, 91
-start 83, 91
-subst 53, 72
-timer 44, 62
-TRAM 63
-turnLeft 66
-turnRight 65
-V 29
-v 28
-vertical 65
-width 46, 67
-xOffset 48, 68
-y 97
-yOffset 49, 69
$ 102
# 102
\??? 93
\ (パスの区切りとしての~) 2, 24
\ (エスケープキャラクタ) 102
\\ 93, 103
\d (deselect) 93
\e (エスケープコード) 93
\f (改ページ) 93
\n (改行) 93
\r (復帰) 93
\s (サブコード) 93
\t (水平タブ) 93
\v (垂直タブ) 93
\x?? 93
%% 93
%?D 91
%?d 91
%?I 92
%?i 92
%?M 91
%?m 91
%O 93

C

cell 領域 45, 58
cfilter 110, 121
cmplain.base 21

D

define 104,117
dfilter 110, 121
dpi 96

E

env 108, 118
error 文 101, 116, 128

F

filter 110, 121
font 122, 133

G

generator 109, 119

I

if 文 101, 114
ifdef 文 114, 124
ifeq 文 115, 126
ifndef 文 115, 125
ifneq 文 116, 127
include 文 101, 113, 123
inimf.x 94
initex.x 24

L

landscape.sty 65, 66
local.mf 21

M

makefont.x 20, 95
MFBASES 18, 21
MFHOME 20, 21
MFINPUTS 18, 21
MFPOOL 19
mf.pool 19
mixer 110, 120

P

p3m ファイル 112, 129
PREVIEW_CFG 9
PREVIEW_P2M 10
PREVIEW_P3M 11
PREVIEW.CFG 9
PREVIEW.P2M 10
PREVIEW.P3M 11
PRINT_CFG 14
PRINT_P2M 15
PRINT_P3M 16, 70
PRINT.CFG 14
print.cfg 55
PRINT.P2M 15
print.p2m 15, 70
PRINT.P3M 16, 70
print.p3m 70
ptex.pool 4

T

TEXEDIT 5
TEXFONTS 4
TEXFORMATS 3
TEXHOME 6, 8, 13
TEXINPUTS 3
TEXPKS 8, 13
TEXPOOL 4

V

virmf.x 94
virtex.x 24

W

warning 文 101, 116, 128

あ

アクセント記号 130
一般定義 130
引数 104
エスケープキャラクタ 102

か

拡大率 96
関係記号 137
記号 134, 139
行継続 102
ギリシャ文字 136
空行 102
コメント 102
コンフィギュレーションファイル . 100
コンフィギュレーションファイル名 26

さ

識別子 103
数学記号 135, 138
制御文 101

た

定義体 104
定義文 100
特殊記号 134
ドライバ定義文 100, 109
ドライバネーム 109, 112

な

二項演算記号 137

は

フォーム 103
フォントエイリアスネーム 112
フォントジェネレータ 109
フォント定義文 100, 112
フォント定義本体 112
フォントドライバ 109
フォントフィルタ 109, 112
フォントミキサ 109, 112
フォント名 96
文 100
ベースライン 143

ま

マクロ定義文 100, 104
マクロネーム 104

や

矢印 136

ら

例外定義 130

わ

割り算ファクタ 92

本書に添付されたディスク内の、プログラム、データおよびそれに準じるもの (以下、これらをまとめて「プログラム」と総称します) は、正しく動作することを望んで製作されていますが、その動作はいっさい保証されていません。したがって、各プログラムを使用したために生じたいかなる損害についても、以下に名前をつらねる人および法人がその補償をすることはありません。各「プログラム」は、ユーザの責任において使用してください。

また、各「プログラム」は、刊行時点の最新版を添付するように努めていますが、さまざまな理由から、必ずしも最新版ではないことがあります。この場合でも、以下で名前をつらねる者に、これに対応する義務はないものとします。

本書の内容に関するお問い合わせは、返信用の封筒に切手を貼ったものを同封のうえ、必ず封書で「ソフトバンク株式会社 ハードウェア活用書編集部 『X680x0 TEX』係」までお願い致します。なお、本書の内容以上に関するお問い合わせにはお答えしかねますので、御了承ください。

X68k Programming Series #3

X680x0 TEX

1994年7月26日　初版 第1刷 発行

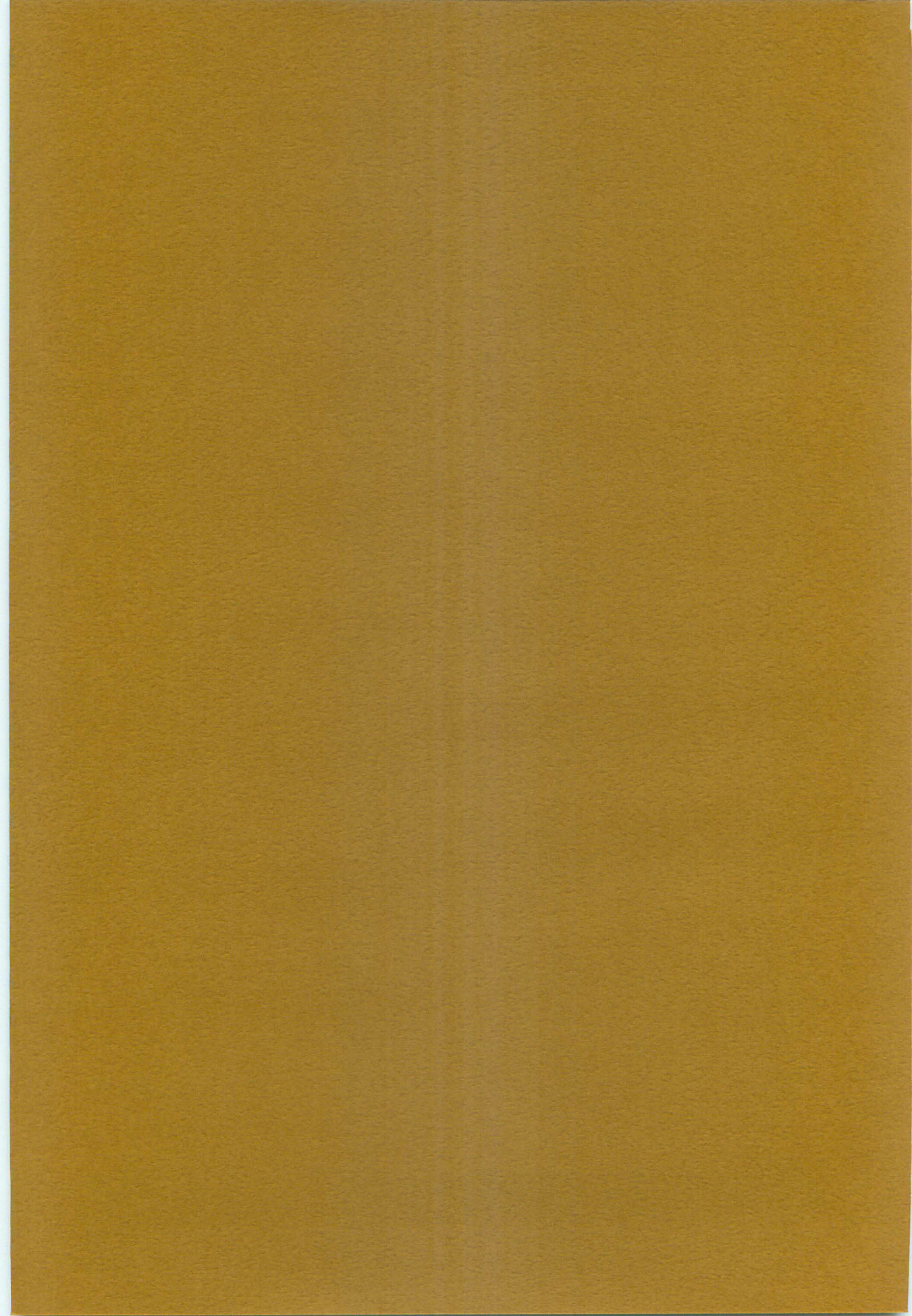
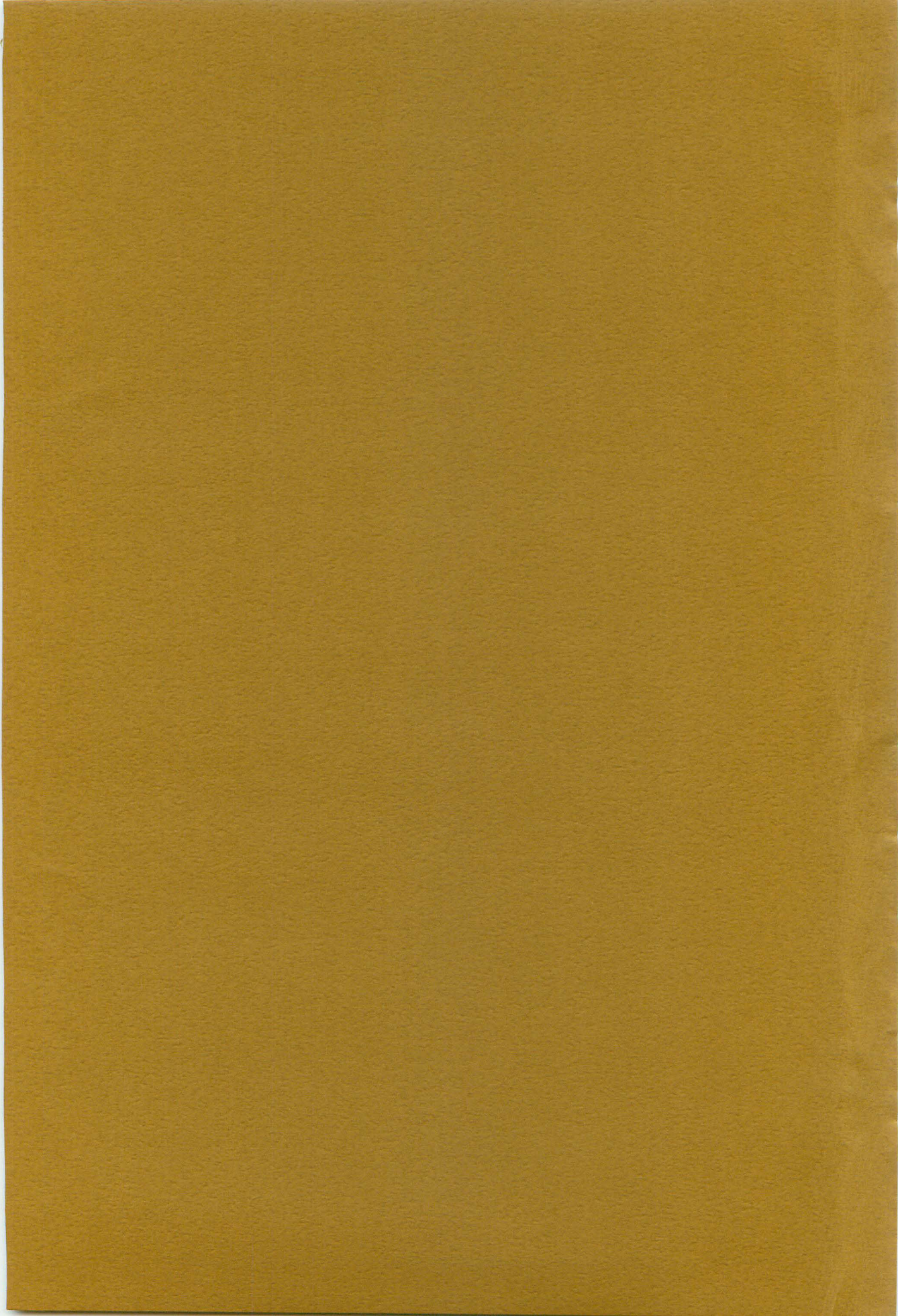
著　者　吉野(よしの) 智興(ちくみ)　川本(かわもと) 琢二(たくじ)
　　　　山崎(やまざき) 岳志(たかし)　実森(じつもり) 仁志(ひとし)

発行者　橋本　五郎

発行所　ソフトバンク株式会社　出版事業部
　　　　〒103 東京都中央区日本橋浜町 3–42–3
　　　　TEL　販売　03 (5642) 8101
　　　　　　　編集　03 (5642) 8140

印　刷　東京書籍印刷株式会社

ISBN 4-89052-542-4

Cover Design = Tetsuya Yonetani
Style Design = Tateaki Hori

ISBN4-89052-542-4

C0055 P9800E

3点セット定価9,800円
(セット本体9,515円　分売不可)

X68k
Programming Series
(#3)
X680x0
TEX